**Queensland University of Technology**

# A Data Mining Framework for Relevance Feature Discovery

by

Luepol Pipanmaekaporn

(B.Eng, M.Sc.)

A dissertation submitted for the degree of

Doctor of Philosophy

School of Electrical Engineering and Computer Science

Faculty of Science and Engineering

# Abstract

Automatic discovery of relevance features in real-world data for describing user information needs or preferences is a new challenge in data mining community. For many years, several research efforts in information retrieval (IR) and information filtering (IF) have attempted to address the difficult issue, using term-based and phrase-based approaches. However, many experiments do not support the effectiveness of using these traditional approaches because there are many redundant and noisy features extracted in the data.

Recently, pattern mining-based approaches to relevance feature discovery (RFD) have been proposed to overcome the quality issue of feature extraction in user relevance feedback. These approaches basically utilise closed sequential patterns in text to improve the quality of features for information filtering (IF). According to experimental results, the data mining-based approaches can achieve encouraging performance in comparing with traditional IF ones. Nevertheless, conventional pattern mining techniques usually give rise to the large output size and often limit their effective use. Furthermore, some discovered patterns that may capture meaningless or include uncertainties can affect the quality of extracted features in describing a specified topic. Thus, it is still an open research issue to guarantee the quality of such features by pattern mining.

This research presents an innovative data mining framework for RFD. This framework efficiently mines a training set, including relevant and non-relevant documents, for closed sequential patterns. We also introduce a new data mining

technique, *pattern cleaning*, to refine the discovered patterns for describing the user's topic. Finally, the utilisation of the discovered knowledge is performed to enhance the quality of extracted features in text. Numerous experiments within information-filtering (IF) domain are conducted on Reuters Corpus Volume 1 and TREC topics. The experimental results confirm that the proposed approach achieves encouraging performance as compared to pattern-based approaches and state-of-the-art IR approaches. This research also contributes new direction to many areas including data mining, machine learning and information retrieval.

# Contents

# List of Figures

# List of Tables

# Abbreviations

| | |
|---|---|
| BoW | Bag-of-words |
| RFD | Relevance feature discovery |
| DP | Discriminative patterns |
| EP | Emerging patterns |
| FP | False positive |
| GR | Growth rate |
| IDF | Inverse document frequency |
| IF | Information filtering |
| IG | Information gain |
| IR | Information retrieval |
| NIST | National Institute of Standards and Technology |
| POS | Part-of-speech |
| PRW | Probabilistic relevance weighting |
| RDF | Relevant document frequency |
| RelDF | Relative document frequency |
| RF | Relevance feedback |
| TF | Term frequency |
| TFIDF | Term frequency inverse document frequency |
| TN | True negative |

| | |
|---|---|
| IAP | Interpolated average precision |
| IPE | Inner pattern evolution |
| MAP | Mean average precision |
| PCM | Pattern cleaning model |
| PDS | Pattern Deploying Support |
| PTM | Pattern taxonomy mining |
| JEP | Jumping emerging patterns |

# Statement of Original Authorship

The work contained in this thesis has not been previously submitted to meet requirements for an award at this or any other higher education institution. To the best of my knowledge and belief, the thesis contains no material previously published or written by another person except where due reference is made.

**Signed:** (LUEPOL PIPANMAEKAPORN)

**Date:** 04/09/2013

# Acknowledgement

This research would not have been possible without the support of many people. First of all, I would like to express my immense gratitude to Professor Yuefeng Li, my principal supervisor, for all his guidance and encouragement throughout this research work. He has been always there providing sufficient support with his excellent expertise in this area. Thanks also go to my associate supervisor, Associate Professor Shlomo Geva, for his generous support on my work during this candidature. I would also like to thank Associate Professor Yue Xu and Dr. Abdulmohsen Algarni for their precious comments and suggestions.

Special thanks go to my wife Anne and my parent for listening and tremendous support over these years. I am indebted to all my colleges who have supported me at QUT's e-Discovery Research Lab, especially group members: Dr. Xujuan Zhou, Dr. Yan Shen, Mubarak Albathan, Bin Liu for offering invaluable advice and discussion regarding my research work. Thanks also go to Dr. Shen-Tang Wu (Sam) for the data pre-processing and the construction of some baseline models. Last but certainly not the least I would like to thank my examiners for their useful comments and suggestions.

# Chapter 1

# Introduction

## 1.1 Motivation

With the explosion of information resources on the web, nowadays users may have increasing difficulty extracting useful information from the huge amount of accessible data sources. There is an imminent need for effective and efficient tools of searching and retrieving information that the users want.

Traditionally, identification of relevant objects (typically documents) is a classical problem in information retrieval (IR) [8, 81]. The aim of an IR system is to retrieve all the objects that are relevant to a user query (i.e., an information need). Several IR models, such as the bag-of-words model [80] and probabilistic models [53, 79]) have been developed to meet the needs of users. However, IR systems generally focus on the development of global retrieval techniques, often neglecting individual user needs and preferences [11].

It is difficult for users to express their information needs. This is because the

users sometimes have no technical knowledge or no idea in describing the topics that they want. For example, one user may search for "AIDS" (Acquired ImmunoDeficiency Syndrome) in order to find its general symptoms and understand its meaning, while the other may wish to find its effective treatment on Zidovuline/therapeutic use. Instead, it is easier for users to answer which of documents (or objects) are relevant or irrelevant. Relevance feedback is an effective technique used widely in the IR area [74, 78, 96]. The objective of using relevance feedback is to find useful information available in a feedback set, including relevant documents and non-relevant documents, for describing the specific needs of users. IR-based techniques for relevance feedback have been proposed to revise a search query in order to improve retrieval performance. For example, Rocchio [46], the popular term-based method, that utilises relevance feedback to build a model of specific needs for users. Another example is Okapi BM25 [75] that is a document ranking function using a term-weighting technique.

Information filtering (IF) [11, 76] is a research area that offers tools to help users searching for relevant information in large volumes of information sources. Unlike IR, the aim of IF is to screen out irrelevant information from incoming steams of information as well as deliver relevant information to users who need it. Over the years IF systems have been developed for various application domains, such as filtering news [52], emails [31], or even multimedia [43]. In IF systems, user needs or preferences are expressed as user profiles [82]. Unfortunately, manual creation of a user profile is difficult and is not effective. Generally, IF systems utilise machine learning techniques to automatically extract the profile of user needs from a feedback set of data. For example, SVM-based approach [25] and

Latent Semantic Index (LSI) [31].

Both IR and IF systems generally use terms extracted from feedback documents as features to represent a document and a specified topic. The main advantage of term-based methods is to obtain efficient systems as well as advance in term-weighting techniques [63, 74]. However, term-based methods often suffer from the problems of polysemy and synonymy [99]. Furthermore, there are many noisy terms extracted from text documents. Such noisy terms has a major impact on the performance of IR and IF systems [7, 51].

For many years, people have held the hypothesis that using phrases for a representation of document and topic should perform better than terms [24, 95]. A phrase refers to a sequence of words appearing in a sentence or a paragraph. The main advantage of phrases over terms is its more context. Both statistical phrases (i.e., n-grams [45, 90]) and syntactic phrases [9, 26] have been used in some IR and IF systems. Nevertheless, the phrase-based methods did not yield statistically significant improve the performance in comparing with term-based ones [87]. The main reason is that many phrases have inferior statistical properties to words. Moreover, there are large number of redundant and noisy phrases among extracted phrases in documents. The key challenge of phrase-based methods in to find useful phrases for a specified topic.

## 1.2 Closed sequential patterns in text

Frequent pattern mining (FPM) is now one of the most important techniques in data mining. FPM plays an important role in extracting useful knowledge for

describing the data [39].

For many years, many efficient algorithms (such as Apriori-like algorithms [2], Prefix-Span [37], and GST [85]) have been proposed to extract different kinds of patterns, including itemsets, sequential patterns [3], graph patterns [5], and tree patterns [19]. A pattern is called a *frequent pattern* if its frequency is greater than a threshold (*min_sup*). However, one serious drawback that has limited the practical use of FPM is that a lot of patterns can be extracted, and most of them are either redundant or noisy patterns. Currently, data mining has developed some techniques (e.g., maximal patterns [10], closed patterns [108, 110], and representative patterns [101]) for removing redundant and noisy patterns.

Among kinds of patterns, closed sequential patterns used in data mining community have been shown certain extent improvements on the effectiveness of text mining [29, 44, 99]. The closed patterns have turned out to be a promising alternative to phrases, and have some desirable properties: (1) they enjoy statistical properties like terms (2) many redundant or meaningless patterns are removed with respect to others, and (3) they are lossless representation for all frequent patterns. The following example explains the concept of closed sequential patterns in text. Figure 1.1 shows a sample of feedback documents from the RCV1 dataset [55]. Table 1.1 illustrates all frequent sequential patterns with $min\_sup \geq 0.2$ (or (20%) of the total number of paragraphs in this document). As shown in Table 1.1, the sequential patterns in the sample document refer to a list of terms that appear in each paragraph in the same order. A sequential pattern $\alpha$ is called a *closed* sequential pattern if there exists no sequential pattern $\beta$ such that $\alpha \sqsubset \beta$ and the frequencies of $\alpha$ and $\beta$ are equal. Table 1.2 shows a

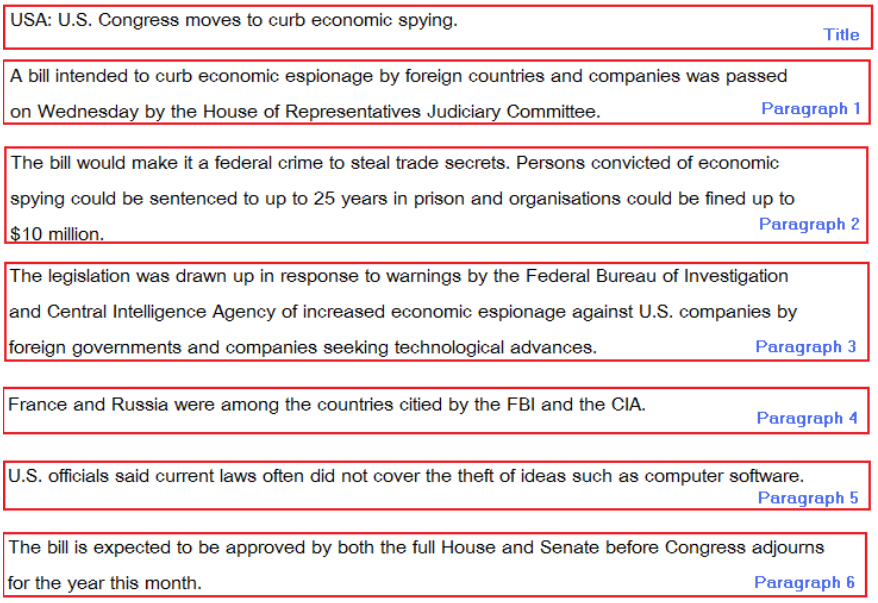



FIGURE 1.1: A sample of a document

list of closed sequential patterns extracted from the sample document in Figure 1.1.

As shown in Table 1.2, the closed patterns are used as high-level features to represent the sample document.

Pattern taxonomy model (PTM) [98] first adopted the concept of closed sequential patterns in text classification. In PTM, all closed sequential patterns and their relationship are extracted as a user profile. However, less significant improvements are made compared with term-based methods. The most likely reason is that many patterns that may contain noise when extracted from the relevant samples. Furthermore, PTM is lacking of effectively using specific long patterns in text.

To overcome the difficulty of using specific long patterns, a deploying method

| **Closed Sequential Patterns** | **Frequency** |
|---|---|
| $[year]$ | 2 |
| $[espionag]$ | 2 |
| $[compani]$ | 2 |
| $[hous]$ | 2 |
| $[econom]$ | 4 |
| $[bill]$ | 3 |
| $[feder]$ | 2 |
| $[curb]$ | 2 |
| $[spy]$ | 2 |
| $[foreign]$ | 2 |
| $[countri]$ | 2 |
| $[congress]$ | 2 |
| $[foreign, compani]$ | 2 |
| $[curb, econom]$ | 2 |
| $[espionag, foreign]$ | 2 |
| $[espionag, compani]$ | 2 |
| $[econom, spy]$ | 2 |
| $[econom, espionag]$ | 2 |
| $[econom, foreign]$ | 2 |
| $[econom, compani]$ | 2 |
| $[feder, econom]$ | 2 |
| $[bill, econom]$ | 2 |
| $[bill, hous]$ | 2 |
| $[bill, year]$ | 2 |
| $[espionag, foreign, compani]$ | 2 |
| $[econom, foreign, compani]$ | 2 |
| $[econom, espionag, foreign]$ | 2 |
| $[econom, espionag, compani]$ | 2 |
| $[econom, espionag, foreign, compani]$ | 2 |

TABLE 1.1: Sequential Patterns extracted from the sample document in Figure 1.1 with $min\_sp \geq 0.2$

[99] for the PTM model has been proposed. This method utilises all the closed patterns in relevance feedback to obtain low-level features for a representation of a document and a topic. The experimental results have shown that the deploying

| **Closed Sequential Patterns** | **Frequency** |
|---|---|
| $[econom]$ | 4 |
| $[bill]$ | 3 |
| $[countri]$ | 2 |
| $[congress]$ | 2 |
| $[bill, econom]$ | 2 |
| $[bill, hous]$ | 2 |
| $[bill, year]$ | 2 |
| $[econom, spy]$ | 2 |
| $[feder, econom]$ | 2 |
| $[curb, year]$ | 2 |
| $[econom, espionag, foreign, compani]$ | 2 |

TABLE 1.2: Closed Sequential Patterns extracted from the sample document in Figure 1.1

method is very effective for using closed sequential patterns in text mining.

Nevertheless, we believe that the quality of discovered knowledge in relevance feedback using the existing pattern mining-based methods has suffered from the two following problems. The first problem is that many closed sequential patterns may contain noise when extracted from relevant documents. For example, short patterns are often common or uninformative patterns. Such patterns can cause the extraction of noisy features that may retrieve information unrelated to user needs. The second problem is that the number of relevant documents in a feedback set is usually limited due to expensive human labelling. This makes the relevant samples to be hardly representative for a specified topic, and then affects the quality of extracted features.

For many years, it is believed that negative relevant information available in non-relevant documents is very useful to help users searching for accurate information

[6, 60, 93]. Most existing methods which use both relevant and non-relevant samples for IF usually focus on revising weights of terms that appear in both the samples. For example, Rocchio [46] and SVM based filtering models [25]. However, using negative relevance feedback in pattern-based approaches to largely improve filtering accuracy is still an open research issue.

## 1.3 Problem statement

Pattern taxonomy models (PTM) with closed sequential patterns [98, 99] have been proposed to overcome the limitation of traditional term-based approaches in IR. However, closed sequential patterns generally focus on only removing noisy patterns with respect to redundancy. As a result, many patterns that may contain noisy information in relevant documents can adversely affect PTM systems. Recently, a pattern-based model for using negative relevance feedback in PTM has been proposed [112]. This approach uses information in non-relevant documents to improve the quality of extracted features. The result shows that using negative relevance feedback has better effectiveness results. However, the improvement gain of the model that uses both feedback is not significant compared to the model using only relevance feedback. The key challenge is how to find useful patterns for discovering specific features in user relevance feedback.

In this thesis, we focus on the problem of mining useful patterns to fulfil user information needs. We propose a new pattern-based model that uses both relevance and non-relevance feedback for relevance feature discovery. This approach adopt the concept of pattern taxonomy with closed sequential patterns [98] to

capture semantics information in a document. We propose an efficient algorithm for pattern taxonomy mining in a large feedback set of documents. Among the discovered patterns, many of them may contain *irrelevant* or *meaningless* information. Such patterns should be considered as noise in describing user information needs. However, removal of all the noisy patterns may adversely affect the quality of extracted features because some of them may include useful information (i.e., terms).

To find useful patterns in text, we introduce the concept of "offenders", a set of non-relevant documents that are closer to relevant ones than others. By using the offenders, noisy patterns can be identified. In this thesis, we propose a new data mining method for mining useful patterns in relevant and non-relevant documents using the offenders, called *pattern cleaning*. We show that the pattern cleaning method has a nice property of anti-monotone pruning. Thus, it is efficient for removing large noisy patterns. Finally, relevance feature models with the concept of pattern deployment are present to use the cleaned knowledge for the effectiveness of relevance feature discovery.

## 1.4 Contributions

The main contributions of this thesis can be listed below:

- We present a pattern mining model for discovering specific features in both relevance and non-relevance feedback to fulfil user information needs.

- We present an efficient algorithm for pattern taxonomy mining that is a nice way for text representation with closed sequential patterns.

- We define a formal definition of noise in context of user information needs and present an efficient method for reducing the effect of noise by pattern mining to improve the quality of extracted features in user relevance feedback.

## 1.5 Publications

- Luepol Pipanmaekaporn and Yuefeng Li. A pattern discovery model for effective text mining. In *Proceeding of the 8th International Conference on Machine Learning and Data Mining (MLDM 2012)*, Springer Lecture Notes in Computer Science, Berlin, Germany, pages 540-554, 2012.

- Luepol Pipanmaekaporn and Yuefeng Li. Discovering Relevant features for effective query formulation. In *Proceeding of the 5th International Retrieval Facility Conference (IRFC 2012)*, Springer Lecture Notes in Computer Science, Vienna, Austria, pages 137-151, 2012.

- Luepol Pipanmaekaporn and Yuefeng Li. Mining a Data Reasoning Model for effective text mining. IEEE Intelligent informatics Bulletin 12(1), pages 17-24, 2011.

- Luepol Pipanmaekaporn, Yuefeng Li and Shlomo Geva. Deploying Top-k Specific Patterns for Relevance Feature Discovery. In *Proceeding of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Conference on Intelligent Agent Technology-Workshops*, Toronto, Canada, pages 318-321, 2010.

## 1.6 Thesis structure

The rest of the thesis is organised as follows:

- **Chapter 2** comprehensively introduces the core concepts of knowledge discovery and frequent pattern mining. In this chapter, it reviews the current work on data mining for finding useful patterns. Finally, the classical and current work on relevance feature discovery in text will be discussed.

- **Chapter 3** provides a theoretical framework for the proposed approach, which encompasses three major components: (1) pattern taxonomy mining with closed sequential patterns; (2) pattern cleaning; and (3) relevance feature models for using the closed patterns in text classification.

- **Chapter 4** provides definitions of pattern taxonomy with closed sequential patterns in text. This chapter also presents a novel efficient algorithm for pattern taxonomy mining, called Pattern Taxonomy Mining (*PTMining*). We describe efficient mechanisms in PTMining to speed up a mining process on a large collection of documents.

- **Chapter 5** gives the formal definition of noise in user relevance feedback, and discusses some issues regarding the quality of discovered knowledge by pattern mining. It also introduces a pattern cleaning method for mining useful patterns in relevant and non-relevant documents. We analyse some theoretical properties in the pattern cleaning method, which makes it to be efficient for pruning many noisy patterns.

- **Chapter 6** explains how to employ the discovered relevant knowledge obtained by the pattern cleaning method. for the effectiveness of information filtering. In this chapter, we propose two relevance ranking models for document filtering: one that directly use the relevant patterns as a new feature space for describing the user relevant documents and another that extract low-level features based on the mined results of patterns to improve the use of specific long patterns in text.

- **Chapter 7** details the benchmark datasets and performance measures, and discusses the application of the proposed pattern-mining models to IF. A detailed analysis of the comparison results of the experiments is also presented in this chapter.

- **Chapter 8** concludes this thesis and draws the direction for future work.

# Chapter 2

# Related Works

In this chapter, the literature related to the particular area of study is reviewed. It is organised according to the following three major topics: (1) knowledge discovery, (2) frequent pattern mining, and (3) relevance feature discovery.

## 2.1 Knowledge Discovery

Several definitions have been given for the term "knowledge discovery" [28, 30]. However, one commonly used definition coined by [28] is that

> "Knowledge Discovery is the non-trivial process of identifying valid, novel, potentially useful and ultimately understandable patterns in data".

According to this definition, the knowledge discovery can be formally defined as follows: Given a set of facts (data) $\mathcal{F}$ and a language $\mathcal{L}$, and some measure of certainty $\mathcal{C}$, a *pattern* is a statement $S \in \mathcal{L}$ that describes relationships among

a subset $\mathcal{F}_s$ of $\mathcal{F}$ with a certainty $c$. A pattern is called *knowledge* if it is interesting and certain enough, according to the user's interests and criteria. As a consequence, knowledge discovery is to extract interesting patterns from a set of facts in a database.

### 2.1.1 Knowledge Discovery Process

Knowledge Discovery in databases (or KDD) is an interactive and iterative process, which usually involves steps, with decisions made by users. Figure 2.1 illustrates the process of KDD as defined in [28]. As seen in Figure 2.1, the steps of KDD consist of data selection, data pre-processing, data transformation, data mining and evaluation . The function of each of these steps is described below.

- **Data Selection:** This step involves generating or selecting a dataset to be performed by the knowledge discovery. The input of this process is a database and output is a target data.

- **Preprocessing:** In this step, basic operations for data cleaning and noise removing are performed. Required information is also collected to model or account for noise, and appropriate strategies are determined for dealing with missing data and accounting for redundant data.

- **Transformation:** The pre-processed data needs to be transformed into a predefined format, depending on the data mining task. This step requires the selection of adequate types of features to represent the data. Feature selection can also be used at this stage for dimension reduction. As the end of this process, a set of features is recognised as a data set

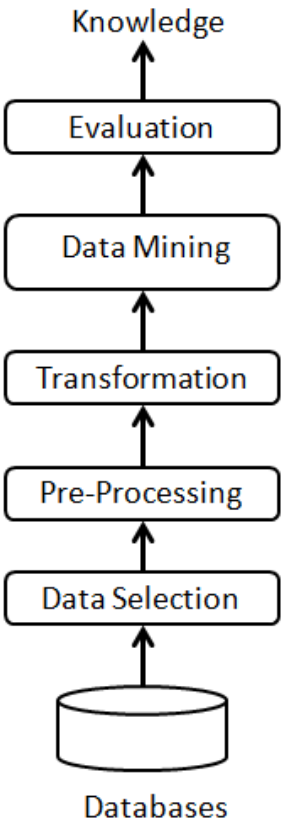



FIGURE 2.1: The KDD process model in [28]

- **Data Mining:** This process involves searching for interesting patterns in a particular representational form or a set of such representations, including classification rules, trees and clustering. The user can aid data mining by correctly performing the preceding steps.

- **Evaluation:** The discovered patterns are evaluated for whether they are valid, novel and potentially useful for the users, to meet their information needs. Only those evaluated to be interesting in some manner are viewed as useful knowledge. This process should decide whether a pattern is interesting enough to form knowledge in the current context.

### 2.1.2 Data Mining Tasks

In the KDD process, the data-mining is the key important step to search for interesting patterns to perform different knowledge tasks. For example, to recommend a product (item) to a particular user, a recommender system might need to discover from a customer database the interesting patterns of users who have similar taste to that particular user. Based on these patterns, a rating can be predicted for products with the user in mind; that is, to what extent will the user like a recommended product. Another predictive use of data mining is to discover patterns in past weather data to forecast weather for the immediate future.

In general, data-mining tasks have one of two main objectives: *prediction* or *description*. Predictive data mining can be defined as the searching for predictive patterns in an available set of data that are useful for forecasting or predicting the behaviour of unknown or new data. Descriptive data mining is the process of finding human-interpretable patterns describing the data.

The main tasks in predictive data mining can be listed as the following:

- **Classification:** Classification is the process of assigning data objects to desired predefined categories or classes. This can be viewed as the process of finding a proper method to distinguish data classes or concepts. Generally, training data is required for concept learning, before classification can proceed.

- **Regression:** Regression refers to cases that focus on the relationship between a dependent variable and one or more independent variables to predict the unknown or future value of other interesting variables.

The following list of common tasks make up descriptive data mining.

- **Clustering:** Given a set of data objects, clustering is the task of partitioning an object set into a finite number of groups such that the objects in the same group have similar characteristics. In other words, the purpose of clustering is to maximise intra-class similarity and minimise inter-class similarity. The major difference between classification and clustering is that the latter analyses objects without consulting class labels, whereas the former needs such information in a supervised setting.

- **Summarization:** This task is to analyse a set of data objects and describe their common or characteristic features. Redundant features are removed to generate a set of compact patterns, representing the concept of these objects.

- **Association Analysis:** Given a set of data objects, the association task is to find implicit relationships between features in the data set (that is, items or attributes) with respect to a given criterion. For example, these relations may be associations between attributes within the data item (intra-patterns) or associations between different data items (inter-patterns).

## 2.2 Data Mining Techniques

Data mining generally utilises methods in the areas of machine learning, statistics, artificial intelligence and databases. Data-mining techniques can be roughly divided into two major categories: *global* and *local*

Global techniques aim to build models that describe the overall tendencies of data for making decisions. Popular global data-mining techniques include decision trees, rule induction algorithms, genetic algorithms, Naive Bayes, K-nearest neighbour and support vector machines (SVMs) [100]. However, these global data-mining techniques have poor interpretability of the mined results, which makes them difficult to examine by users. Furthermore, most of those techniques try to learn approximate concepts to avoid the cost of training time. As a consequence, the accuracy of data mining models with these techniques is often degraded [39].

In contrast, local mining techniques try to find interesting patterns that describe particular parts of the data rather than modelling the entire dataset. Such patterns represent local structures (relationships) of entities (for example, items, features or terms) in the data. The main advantage of data mining with local techniques is that it is interpretable. Furthermore, many efficient mining techniques have been developed to find patterns in very large databases.

## 2.3 Frequent Pattern Mining in Knowledge Discovery

Frequent pattern mining (FPM) is one of the core techniques in data mining. FPM plays an important role in discovering interesting patterns that represent local structures of entities (for example, items, terms, or other objects) in the data [62]. Starting from association-rule mining [2], many studies have been conducted on FPM, ranging from scalable pattern-mining algorithms to various domain applications [39].

FPM also encompasses several techniques for finding various kinds of patterns like itemsets [4], sequences [85], trees [19], and graphs [114]. The general goal of FPM is to find all patterns with a frequency no less than a minimum frequency threshold in a given database.

Below, we briefly review the common tasks of pattern mining.

- **Itemset Mining** Itemset mining has been extensively studied in pattern mining. The problem of itemset mining is to discover all frequent sets of entities (or items or attributes). Since its introduction by [4], many efficient algorithms and interesting methods have been proposed for itemset mining [39].

- **Sequence Mining** Sequential pattern mining is concerned with finding all frequent sequential patterns across time (or position) in a sequence database [85]. The popular efficient algorithms for mining sequential patterns include PrefixSpan [37] and SPADE [109].

- **Tree Mining** Given a database of tree objects, the goal of tree mining is to find all of the commonly occurring frequent sub-tree patterns. Several algorithms have been proposed recently for tree mining, starting with the work of [19].

- **Graph Mining** Another important pattern-mining problem is to find all frequent sub-graph patterns in a database of graphs. There are many applications involving graph data such as social network analysis, protein structure and chemical compounds. Some of the early works in graph mining include [65, 114]. In addition, many recent methods have been proposed to improve the efficiency of mining graph patterns such as *gPrune* [114] and Uniform Sampling method [5].

### 2.3.1 FPM Preliminaries

In this section, we briefly review the formal definitions of frequent pattern mining.

- A **database** $\mathcal{D}$ is a collection of transactions. A transaction $t_i \in D$ may be also called an *object*, an *event*, or a *record*. Let $\mathcal{T} = \{t_1, t_2, \ldots, t_n\}$, be a set of $n$ distinct transaction identifiers, written as *tid*. With the *tid*, an event can be represented as a tuple, $\langle t, \mathcal{E} \rangle$, where $t$ is denoted as the *tid* and $\mathcal{E}$ is the corresponding event. The number of transactions in $\mathcal{D}$ is defined as $|\mathcal{D}|$.

- **Pattern** A *pattern* can be generally defined as a set of items or objects etc. Let $I = \{i_1, i_2, \ldots, i_m\}$ be a set of $m$ distinct items in a given database

$D$. A pattern $\phi$ is a collection of items defined according to a language of pattens $\mathcal{L}$. A pattern $\alpha$ is a *sub-pattern* of pattern $\beta$ if $\alpha \subseteq \beta$. If $\alpha$ is a sub-pattern of $\beta$, $\beta$ is a *super-pattern* of $\alpha$.

- **Coverage** A transaction $t_i \in \mathcal{T}$ is said to *contain* or *support* a pattern $\phi$, written as $\phi \subseteq t_k$, if for all items $i_j \in \phi$ it holds that $i_j \in t_k$. For a given database $\mathcal{D}$, it is defined that $coverage(\phi, D) = \{t | t \in \mathcal{D}, \phi \subseteq t\}$, the set of transactions in $D$ that $\phi$ covers;

- **Support** The support of pattern $\phi$ in a database $\mathcal{D}$, denoted as $Sup(\phi, D) = |coverage(\phi, D)| = |\{t \in \mathcal{D} | \phi \subseteq t\}|$.

The problem of frequent pattern mining is that of finding a theory $\mathcal{T}\langle(\mathcal{L}, \mathcal{D}) = \{\phi \in \mathcal{L} | sup(\phi, D) \geq min\_sup\}$, where $min\_sup$ is a predefined threshold, known as a *minimum support threshold*.

### 2.3.2 Pattern Mining approaches

While there are a variety of pattern mining algorithms proposed in data mining, they do share some common approaches to efficiently explore the search space of frequent patterns.

#### 2.3.2.1 Generation-and-Test Approach

Some pattern-mining algorithms follow a *generation-and-test* approach [4, 85]. This approach basically generates one or more candidate frequent patterns per iteration, which are then examined against the database to test whether their

support is greater than the minimum support threshold (*min_sup*). If the test succeeds, the pattern is made a frequent pattern and stored for further generations; otherwise, it is discarded. The main advantage of this mining approach is that if a candidate pattern is pruned, all super-patterns of that pattern are discarded because they cannot be frequent. As the iteration continues, frequent patterns of increasing length are discovered, starting from size one. The process stops when all frequent patterns existing in the database have been discovered. The following describes each of the sub-tasks that comprise the generation-and-test approach.

- **Candidate Generation** The objective of this step is to obtain candidate frequent patterns of size $(k+1)$ from frequent ones of size $k$ where at least two size $k$ patterns are joined to explore size $(k+1)$ candidate ones. In order to reduce the search space, this step uses the *anti-monotone* property, which states that a sub-pattern of a frequent pattern is always frequent and a super-pattern of an infrequent pattern is always infrequent. Consequently, candidates are only generated from the frequent patterns, and all other patterns are pruned from the search space.

  Generally, there are two common strategies for exploring the candidates. The first strategy is a *depth first strategy*. In this strategy, a frequent pattern is extended repeatedly until it cannot be extended any more. The second one is a *breadth first strategy* which generates all possible size $(k+1)$ candidates after all size $k$ frequent patterns are discovered. Well-known

pattern-mining algorithms with a breadth-first strategy include Apriori algorithm for itemset mining [2] and the GSP algorithm for sequential pattern mining [85].

- **Support Counting** The support-counting step is used to compute the support of a candidate frequent pattern in a given database. This step can be considered as the overhead of pattern mining because it requires a full database scan to count occurrences in the database, especially when the database is very large.

#### 2.3.2.2 Compression-based Approach

One disadvantage of the generation-and-test approach is the overhead required in the candidate-generation step. This overhead not only increases the time complexity, but may also be unable to mine the complete set of frequent patterns in a long transaction database because of the exponential combinations of candidate frequent patterns.

Another common approach to frequent pattern mining is the *compression-based* approach. This approach basically avoids the candidate generation step by using a prefix tree, which enables the database to be compressed to fit the main memory. One of the most well known algorithms based on the compression-based approach is *frequent pattern growth (FP-growth)* [38]. The FP-growth algorithm requires only two database scans to find all frequent itemsets The first round is to discover the frequent itemsets of size-1 with minimum support, and the second is to sort the frequent items in all transactions in descending order of support values for

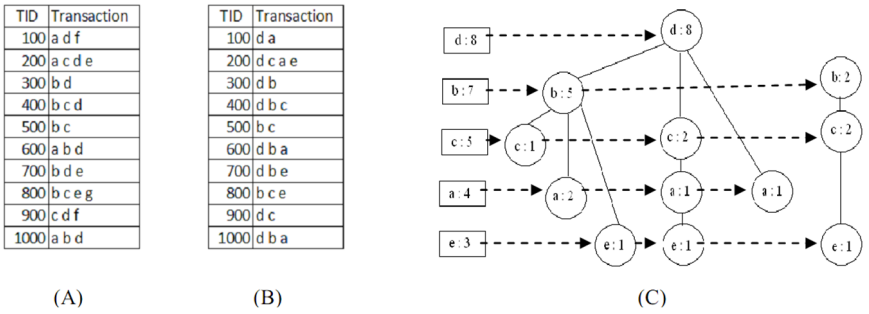



FIGURE 2.2: An example of FP-growth algorithm

extracting the frequent pattern tree. An example of a frequent pattern tree can be seen in Figure 2.2 Once the frequent pattern tree is built, the extraction of frequent itemsets are directly performed on the built tree.

The FP-growth algorithm achieves excellent performance in pattern mining compared to the generation-and-test algorithm. However, it has a drawback in that the frequent pattern tree may not fit memory, especially for large databases. Further, the tree-based algorithm does not allow mining of different kinds of patterns, such as sequential patterns containing ordered sets of items.

One common strategy for mining sequential patterns borrowing the compression-based approach is to use the divide-and-conquer strategy, also called the projection-based technique [36, 37]. The aim of the divide-and-conquer method is to break a large sequence database into a smaller set of sequence databases, called projected databases, and then mine frequent sub-sequences in each projected database For example, *FreeSpan* [36] uses frequent items to recursively project sequence databases into fragments and grow sub-sequence fragments in each projected

database. In PrefixSpan [37], the main idea is to examine only the prefix sub-sequences and project only their corresponding postfix sub-sequences into projected databases. In each projected database, sequential patterns are extended by exploring only local frequent patterns.

The main advantages of these pattern-growth techniques are that the candidate-generation step is avoided and efficiency performance is obtained by using the divide-and-conquer approach.

#### 2.3.2.3 Other Algorithms

Typical pattern-mining algorithms mine frequent patterns in horizontal data format databases with the assumption that each transaction is short or moderate. However, some domain databases are relatively small, with very long transactions. In these cases, such algorithms may not be scalable because of the combinatorial search space.

Some data-mining algorithms have been proposed to solve the scalability of the mining algorithms by exploring the search space in a different way. For example, Carpenter algorithm [67] that explores the search space of frequent patterns in a long transaction database by using bottom-up search strategy. *Eclat* (Equivalence CLAss Transaction) algorithm [111] mines frequent patterns in *vertical data format*, where each transaction in the database consists of a list of transaction ids. Table 2.1 illustrates the vertical format representation of the database in Figure 2.2 (a).

As shown in Table 2.1, Eclat explores the search space by intersecting the transaction-id lists between items based on the Apriori-like mining framework, where frequent

| *Item* | $TID-$**list** | $Sup_a$ |
|---|---|---|
| a | 100 200 600 1000 | 4 |
| b | 300 400 500 600 700 800 | 6 |
| c | 200 400 500 600 800 900 | 6 |
| d | 100 200 300 400 600 700 900 1000 | 8 |
| e | 200 700 800 | 3 |
| f | 100 900 | 2 |
| g | 800 | 1 |

TABLE 2.1: The vertical data format of the database in Figure 2.2 (a)

$(k+1)$ itemsets are identified by the intersected set of transaction ids of frequent $k$ itemsets. For example, given $min\_sup = 3$, the frequent itemset $bd$ can be obtained by intersecting the transaction-id lists of item $b$ and $d$.

## 2.4 Mining useful frequent patterns

A major challenge of pattern mining is *pattern explosion*. Pattern-mining algorithms typically produce a large number of discovered patterns. The large output size of patterns poses problems not only for efficiency, but also hinders the practical use or analysis of the pattern mining results [17].

To overcome the pattern explosion problem, several data mining techniques for mining useful patterns have been developed to enhance the quality of frequent patterns to solve different data mining problems. Generally, they can be defined as either *descriptive* or *predictive*. Figure 2.3 illustrates the taxonomy of post-mining approaches.

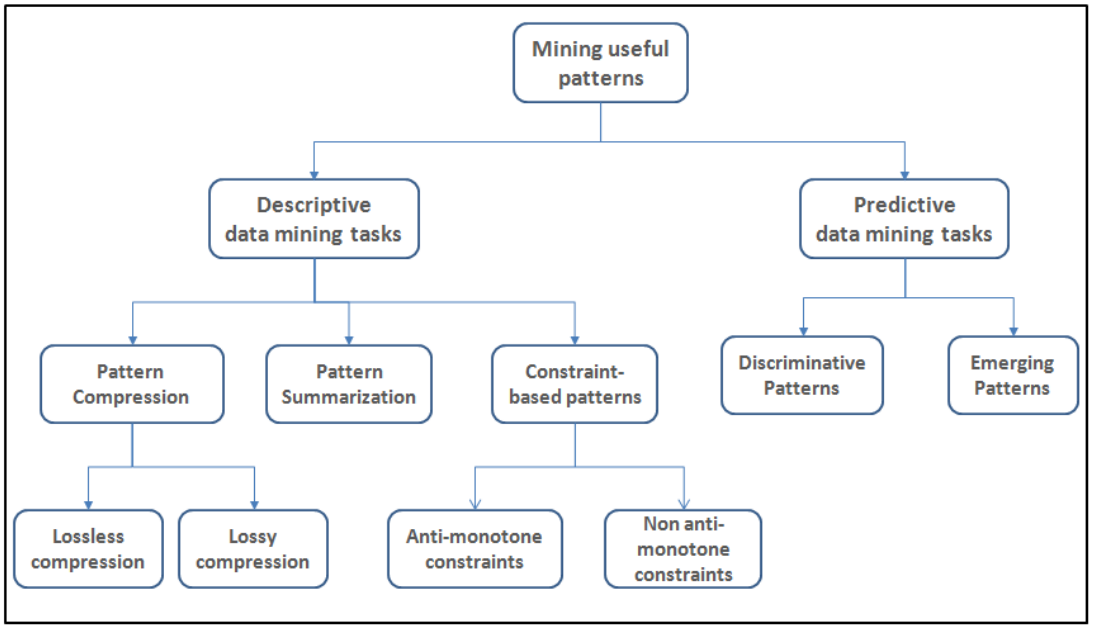



FIGURE 2.3: Mining useful Patterns

### 2.4.1 Descriptive-based Methods

Traditional pattern-mining algorithms typically generate an overwhelming number of discovered patterns, making it difficult to interpret the mined results. The objective of descriptive pattern mining is to identify a subset of patterns that are human interpretable. Popular descriptive pattern-mining techniques include constraint-based pattern mining, pattern compression and pattern summarisation.

#### 2.4.1.1 Constraint-based Pattern Mining

In constraint-based pattern mining, all the patterns that satisfy one or more user-specified constraints are considered as *interesting* patterns. A constraint can be

defined either a quantitative measure (also called an interestingness measure) or a Boolean expression.

There are many constraints proposed in data mining to search for interesting patterns. In other words, they filter out non-interesting patterns. Roughly, these constraints can be grouped into *anti-monotonic* or *non anti-monotonic*. A constraint is called anti-monotonic if it can prune all super-patterns of a pattern that violates the constraint [64]. In other words, if a pattern fails to satisfy the constraint, all super-patterns of the pattern will also fail to satisfy the constraint. This kind of constraint offers the desirable property of reducing the combinatorial search space by pushing them into the mining process. A well-known example of an anti-monotonic constraint is the downward closure used by existing algorithms for mining-association rules (for example, the Apriori algorithm [2]), where if a pattern is *infrequent*, the same is true for all its supersets. Another example of anti-monotone pruning is the *all-confidence* measure [66], which quantifies the strong relationship of items contained in an itemset to reduce the pattern space. In contrast, the majority of constraints for mining desired patterns in data mining are non-anti-monotonic and thus allow for the capture of a different view of interestingness. Examples include *chi-squared test* [84], *lift* [12], and entropy gain *tan2002selecting*. However, these constraints do not possess the downward closure property. Thus, we have no luxury pruning of patterns to reduce the large search space.

The main drawback of constraint-based methods is that, to find desired patterns, users are expected to provide a high-level vision of the data-mining methods, which is not obvious. Further, the question of how to manage multiple constraints

without any conflict remains [69, 86].

#### 2.4.1.2 Pattern Compression

Unlike constraint-based approaches, pattern compression has attempted to overcome the problem of large output size of FPM algorithms. The main idea of pattern compression is to remove redundant patterns with respect to other ones. The previous efforts that have emerged from pattern compression can be roughly divided into two categories: *lossy* and *lossless compression*.

- **Lossy Compression**: The earliest efforts on pattern compression looked at discovering a maximal set of patterns [10]. A pattern is *maximal* if it is not a sub-pattern of any other ones. Depending on the dataset, a maximal set can substantially reduce the complete set of patterns, especially in dense datasets that contain a high level of similarity among transactions. Further, maximal patterns can be efficiently mined during the discovery process. As a compact but expressive representation, maximal patterns are often used to represent the content of a document for solving text-mining tasks, such as in text summarisation [54], document clustering [41].

  The main drawback of maximal patterns is that the compression can give rise to lost support information of the non-maximal patterns. However, efficient algorithms for mining maximal patterns do currently exist, such as those proposed by MaxMiner [10], GenMax [33] and MAFIA [15].

  Recently, the problem of pattern compression has been considered as a clustering problem [101]. From this perspective, a distance function that

computes a distance between frequent patterns is adopted. Two patterns are grouped together if they have a distance less than $\alpha$ threshold. Then, a set of representative patterns for each cluster is extracted as a cluster centroid. As a typical clustering problem is NP-hard, two greedy search algorithms, named *RPlocal* and *RPglobal*, are applied to find an approximate set of representative patterns. The main advantage of the cluster-based approach is that it can apply to different kinds of patterns. The drawback is that a user needs to choose carefully the right value for $\alpha$, as this is not obvious.

- **Lossless Compression**: To overcome the support information loss problem of maximal patterns, the concept of closed pattern mining was first proposed in [68, 108] for association-rule mining. A frequent pattern is *closed* if it has no super-pattern with the same support. The main advantage of mining closed patterns is that all of the frequent patterns and their support information can be retrieved without consulting the database. Further, closed frequent patterns can also be mined efficiently without using a post-processing technique. The compression ratio of closed patterns is smaller than for maximal pattern mining.

  Figure 2.4 illustrates the comparison results of closed patterns and maximal patterns of the database in Table 2.2 when the minimum threshold is set to 2 ($min\_sup = 2$). In this figure, a close-curve encloses the frequent patterns that have the same support. For example, the frequent patterns

$AB$, $ABC$, $ABE$, $ACE$, $AE$, and $ABCE$ can be enclosed together because all the patterns have the same support. According to the concept of closed pattern, pattern $ABCE$ is considered as the closed pattern for the corresponding enclosed patterns since there is no super-pattern of the pattern. Furthermore, pattern $ABCE$ is the maximal pattern for the frequent patterns because it contains its sub-patterns corresponding to the frequent ones.

| $Transaction\ ID$ | $Transaction$ |
|---|---|
| $T_1$ | $A, C, D$ |
| $T_2$ | $B, C, E$ |
| $T_3$ | $A, B, C, E$ |
| $T_4$ | $B, E$ |
| $T_5$ | $A, B, C, E$ |
| $T_6$ | $B, C, E$ |

TABLE 2.2: A transaction database

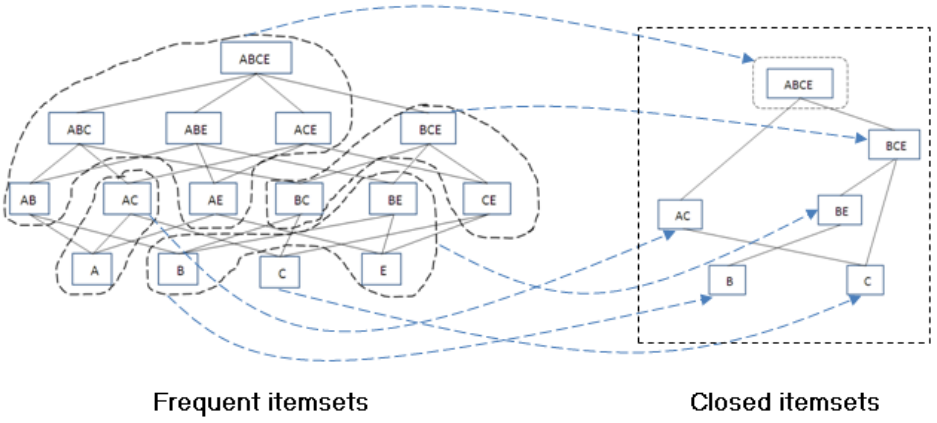



FIGURE 2.4: Closed pattern and Maximal pattern for the frequent patterns from Table 2.2 at $min\_sup = 2$

Compared to maximal pattern mining which allows pruning the large pattern space without taking into account the pattern support information, closed pattern mining focuses on retaining the support information about all patterns. Thus, closed pattern mining is typically more popular. Over the years, several efficient methods for mining closed patterns have been proposed. For example, ones of the well known algorithms for closed itemset mining include CHARM [110] and CLOSET+ [92].

Mining non-derivable itemsets [16] is another well-known lossless compression technique for frequent itemsets. This technique is based on the inclusion-exclusion principal, which enables us to find the lower and upper bound on the support of an itemset based on the support of its subsets. In this way, only a set of non-derivable itemsets, which can be used to derive the support of any derivable itemset, are extracted.

The main limitation of the compression method is that it can only be applied for itemset patterns. Further, for some datasets, the number of non-desirable frequent patterns can be larger than the number of closed frequent ones.

#### 2.4.1.3 Pattern Summarization

Although closed patterns and maximal patterns can largely reduce the number of original patterns, their number may be large depending on the dataset. Recently, some studies have focused on improving the compressibility by finding the best k-set of patterns, as these summarise the original frequent patterns.

The problem of mining the best k-set of patterns for a given collection was first studied in [1]. Since this problem is NP-hard, the authors proposed a greedy algorithm to find the approximate k-set of patterns that cover the whole collection of frequent itemsets with the minimum false positive rate. In [106], the concept of pattern profiles (or master patterns) was introduced to summarize the original set of frequent patterns. A master pattern $M$ is defined as: $< \mathcal{P}, \phi, \sigma >$, where $\mathcal{P}$ is a probability distribution of items in the pattern, $\phi$ is a set of items, and $\sigma$ is the support. Two master patterns $M_\alpha$ and $M_\beta$ are merged into a single one if they have the smallest distance. To generate the k-set of master patterns, k-mean clustering algorithm is used, where k is the user-specified number of master patterns.

The main advantage of the profile-based approach is that it allows for lossless compression, which means that the support information of all patterns can be derived from the k-pattern set. The concept of pattern profiles has also been extended in some studies [89].

One disadvantage of the pattern summarization methods is that they often produce non-interesting patterns due to the ignorance of pattern significance. some studies have proposed the combination of both the significance and redundancy criteria to search for the top k-set of high-quality patterns with minimum redundancy. For example, two greedy search algorithms, MMS and MAS, were developed in [102] to select the top k-patterns for a given collection in terms of both significance and redundancy. A number of quality measures have been proposed for finding top-k interesting patterns regarding the redundancy of patterns [50].

### 2.4.2 Predictive-based Methods

One of the most common tasks of data mining involve prediction. Generally, the task of predictive pattern mining is to discover useful patterns to predict the behaviour of unknown or future data. A special task of predictive data mining is classification, which is defined as the task of prediction of the class label of an object according to a history set of objects. Early studies have focused on mining association rules for use in classification, known as *associative classifiers* [35, 57, 91].

Recently, the focus was more on finding discriminative patterns, called *relevant patterns*, in a training set of objects with class labels. Mining relevant patterns has been shown the positive impact on improving classification accuracy [17, 72], and have been studied under different names, such as emerging patterns [22], relevant patterns [48], and discriminative patterns [17].

Although there have been several algorithms proposed for mining relevant patterns, there algorithms basically use a quality measure for the relevance between a pattern $p$ and a class $c$ of interest. For example, $\chi^2$, F-score, and information gain. After that, the top-ranked relevant patterns are selected. Popular relevance measures include *Confidence* $p(c|x)$ used in associative classifiers [57, 115], *Growth rate*$GR(x, c) = \frac{p(c|x)}{1-p(c|x)}$ used in emerging patterns, and Information gain [17], where $p(c|x)$ denotes the support of a given pattern $x$ for class $c$.

## 2.5 Relevance Feature Discovery

Relevance feature discovery (RFD) is a classical, but challenging task in IR and text mining. The objective of RFD is to find useful features in user relevance feedback (typically text documents) to fulfil user information needs [60]. Traditionally, relevance feedback has been used widely in the area of IR to improve search quality corresponding a given query.

It has been currently used in text mining systems. For example, information filtering (IF) [58, 59] and text classification [71, 83]. Relevance feedback is a subset of retrieved documents that have been judged by the users. The judgement can be 1 which means it is related to the user's topic of interest or 0 which means it is not related to what users want.

### 2.5.1 Traditional term-based approaches

Most existing IR and text mining methods adopted term-based approaches, which describe the text with a vector of terms or keywords associated with their weight. This is known as the bag-of-words (BoW) model [79]. Figure 2.5 shows an example of a BoW representation.

As shown in Figure 2.5, each word in the document is collected in a vector of terms associated with their weight (frequency). BoW representation allows for efficient computation performance and mature techniques for term weighting [63, 83].

To clarify the significance of a term, the frequency with which the term appears in a training set of relevant and non-relevant documents can be used. Each relevant document contains varying amounts of information relevant to a user query.

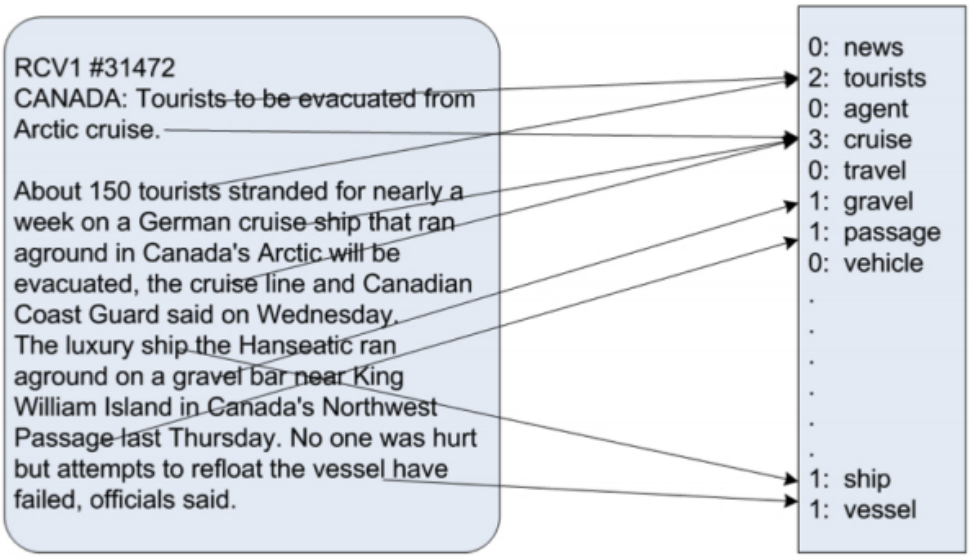



FIGURE 2.5: Bag-of-Words representation using word frequency

The following list of notions will be used to describe the weight function of a given term $t$.

- $r$ is the number of relevant documents that contain term $t$.
- $n$ is the total number of documents in the collection that contain term $t$.
- $R$ is the total number of relevant documents.
- $N$ is the total number of documents in the collection.

Below, some popular term weighting methods in IR are introduced.

- **Term Frequency (TF)**: The frequency of a term $t$ in a document $d$, $TF(d,t)$, can be used as a measure of a term's significance within the document.

- **Inverse Document Frequency (IDF)**: IDF is often used to measure the specificity of terms in a document collection. The assumption is that a term which occurs in many documents in the collection is not a good discriminator and should be given less weight than one that occurs in only a few documents. The formula of IDF can be expressed as:

$$IDF(t) = \log \frac{N}{n} \tag{2.1}$$

- **Term Frequency Inverse Document Frequency (TFIDF)**: The well-known measure used in IR and text mining. TFIDF is the combination of term frequency (TF) and inverse document frequency (IDF).

$$TFIDF(t) = TF(d,t) \times IDF(t) \tag{2.2}$$

- **Probabilistic Relevance Weighting (PRW)**: according to [74], four probabilistic weighting methods for relevant terms were proposed based on the *binary independence retrieval model* [74]. The weighting methods can be listed as:

$$F_1(t) = \log \frac{\left(\frac{r}{R}\right)}{\left(\frac{n}{N}\right)} \tag{2.3}$$

$$F_2(t) = \log \frac{\left(\frac{r}{R}\right)}{\left(\frac{n-r}{N-R}\right)} \tag{2.4}$$

$$F_3(t) = \log \frac{\left(\frac{r}{R-r}\right)}{\left(\frac{n}{N-n}\right)} \tag{2.5}$$

$$F_4(t) = \log \frac{\left(\frac{r}{R-r}\right)}{\left(\frac{n-r}{N-n-R+r}\right)} \tag{2.6}$$

- **Okapi BM25**: The BM25 function was proposed in [75] and aims to weight a term based on its frequency and the document length. The weighting function can be expressed as:

$$W(t) = \frac{tf \cdot (k_1 + 1)}{k_1 \cdot ((1-b) + b\frac{DL}{AVDL}) + tf} \cdot \log \frac{\frac{(r+0.5)}{(n-r+0.5)}}{\frac{(R-r+0.5)}{(N-n-R+r+0.5)}} \tag{2.7}$$

  where $N$ is the total number of documents in the training set; $R$ is the number of positive documents in the training set; $n$ is the number of documents which contain term $t$; $r$ is the number of positive documents which contain term $t$; $tf$ is the term frequency; $DL$ and $AVDL$ are the document length and average document length, respectively; and $k_1$ and $b$ are the experimental parameters.

Below, some popular term weighting methods used in text mining [63, 107] are described.

- **Relevant Document Frequency (RDF)** [47]:RDF has proven effective in text categorisation and filtering. The assumption is that these terms are more specific to a specific collection of documents than terms that occur less. The function can be written as:

$$RDF(t) = r \tag{2.8}$$

- **Relative Document Frequency (RelDF)**: RelDF is an important metric for terms within user-specified documents and general document collections [97]. The idea behind RelDF is to ensure that specific or technical terms that are rare in general usage have a high weighting in text mining. This is expressed as:

$$RelDF(t) = \frac{r}{R} - \frac{n}{N} \tag{2.9}$$

- **Information Gain (IG)**: IG is an information-theoretic metric that measures the difference in the entropy of category prediction by knowing the presence or absence of a term in a document. The formula is written as:

$$IG = -\frac{R}{N}\log\frac{R}{N} + \frac{r}{N}\log\frac{r}{N} + \frac{R-r}{N}\log\frac{R-r}{N} \tag{2.10}$$

- **Mutual Information (MI)**: MI is another metric derived from information theory. In the context of text mining, this metric is commonly applied for measuring the association between a term and a specific document collection; expressed as:

$$MI = log\frac{r/R}{n/N} = log\frac{r}{R} - log\frac{n}{N} \tag{2.11}$$

- **Chi-Square**: Chi-square ($\chi^2$) estimates the difference between observed frequencies and expected frequencies under the independence assumption. It can be applied for measuring the lack of independence between a term

and a topic category. The formula is written as:

$$\chi^2 = \frac{N.(rN - nR)^2}{R.n.(N - R).(N - n)} \tag{2.12}$$

The main advantage of extracting low-level features is to obtain efficient systems and mature in term-weighting techniques. However, the term-based approaches often suffer from the problems of *polysemy* and *synonymy* [112], where polysemy means a word has multiple meanings, and synonymy is multiple words having the same meaning. The semantic meaning of many discovered terms is uncertain for describing user information needs. Furthermore, many noisy terms that are unrelated to the document's main topic extracted from a text document [51]. Such terms may adversely affect IR and text mining systems because irrelevant information can be retrieved by these terms.

### 2.5.2 Pattern Taxonomy Model

Recently, a promising technique for RFD has shifted to data mining. Pattern taxonomy models (PTM) [98, 99, 112] that introduced data mining techniques to information filtering (IF). These approaches basically discover closed sequential patterns in text documents to capture semantic information of a text document. A pattern refers to a list of terms that frequently appeared in a sentence or a paragraph. In the following subsections, the basic definitions in PTM are provided to readers. These definitions are also used in this research work.

In PTM, all documents are split into paragraphs. So, a given document $d$ yields a set of paragraphs $PS(d)$. Let $D$ be a set of feedback documents, which consists

of a set of relevant documents, $D^+$; and a set of non-relevant documents, $D^-$ with respect to the user's judgement.

#### 2.5.2.1 Sequential Pattern Mining

Let $\mathcal{T} = \{t_1, t_2, \ldots, t_m\}$ be a set of terms (or keywords) which can be extracted from the set of relevant documents, $D^+$. A *sequence* $S =< s_1, s_2, \ldots, s_n >$ ($s_i \in \mathcal{T}$) is an ordered list of terms.

*Definition* 1 (**sub-sequence and super-sequence**). A sequence $\alpha =< a_1, a_2, \ldots, a_n >$ is called *sub-sequence* of another sequence $\beta =< b_1, b_2 \ldots, b_m >$, denoted by denoted as $\alpha \sqsubseteq \beta$ but $\alpha \neq \beta$, if there exist integers $1 \leq i_1 \leq i_2 < \cdots < i_n \leq m$, such that $a_1 = b_{i1}, a_2 = b_{i2}, \ldots, a_n = b_{in}$. In addition, we can also say sequence $\beta$ is a *super-sequence* of sequence $\alpha$.

For instance, sequence $< s_1, s_3 >$ is a sub-sequence of sequence $< s_1, s_2, s_3 >$. However, $< s_2, s_1 >$ is not a sub-sequence of sequence $s_1 s_2 s_3$ since the order of item is considered.

*Definition* 2 (**absolute and relative support**). Given a document $d = \{S_1, S_2, \ldots, S_n\}$, where $S_i$ is a sequence of terms contained in a paragraph in $d$. Thus, $|d|$ is the number of paragraphs in document $d$. Let $\alpha$ be a sequence. The *absolute support* of $\alpha$ is the number of occurrences of $\alpha$ in the sequences $S_i \in d$, denoted as:

$$Sup_a(\alpha) = |\{S_i | S_i \in d, \alpha \sqsubseteq S\}|$$

The *relative support* of $\alpha$ is the fraction of paragraphs that contain $\alpha$ in document $d$, denoted as

$$Sup_r(\alpha) = Sup_a(\alpha)/|D|$$

For example, the sequential pattern $\alpha =< t_1, t_2, t_3 >$ in the database, as shown in Table 2.3, has $Sup_a(\alpha) = 2$ and $Sup_r(\alpha) = 0.5$. All sequential patterns in Table 2.4 with absolute support grater than or equal to 2 are presented in Table 2.4.

| **Paragraph ID** | **Sequence** |
|---|---|
| $dp_1$ | $S_1 :< t_1, t_2, t_3, t_4 >$ |
| $dp_2$ | $S_2 :< t_2, t_4, t_5, t_3 >$ |
| $dp_3$ | $S_3 :< t_3, t_6, t_1 >$ |
| $dp_4$ | $S_4 :< t_5, t_1, t_2, t_7, t_3 >$ |

TABLE 2.3: A set of paragraph sequences in a document $d$

| **Sequential Patterns** | $Sup\_a$ | $Sup\_r$ |
|---|---|---|
| $< t_4 >,< t_5 >,< t_1, t_3 >,< t_2, t_4 >,< t_5, t_3 >,< t_1, t_2, t_3 >$ | 2 | 0.5 |
| $< t_1 >,< t_2 >,< t_2, t_3 >$ | 3 | 0.75 |
| $< t_3 >$ | 4 | 1.0 |

TABLE 2.4: All sequential patterns discovered in the sample document in Table 2.3 with absolute support greater than or equal to 2

A sequential pattern $\alpha$ is called a *frequent sequential pattern* if $Sup_r(\alpha)$ or $Sup_a(\alpha)$ is greater than or equal to a minimum support, *min_sup*. Therefore, the problem of sequential pattern mining is to find a complete set of frequent sequential patterns whose support is greater than or equal to a threshold *min_sup*. For example, let $min\_sup = 0.75$, the complete set of sequential patterns which holds

the threshold value include four sequential patterns: $< t_1 >$, $< t_2 >$, $< t_2, t_3 >$, and $< t_3 >$ in Table 2.4.

#### 2.5.2.2 Pattern Taxonomy

It is not uncommon that we can obtain numerous discovered sequential patterns in a text document which may include many redundant or meaningless patterns. Such patterns not only increase both time and space complexities, but may challenge the effectiveness of text mining.

The PTM adopts the concept of closed patterns mentioned in Section 2.4.1.2 as meaningful patterns to represent the semantic information in a text document or a topic.

The formal definition of closed sequential patterns in text is given as follows:

*Definition* 3 (**Closed Sequential Pattern**). A sequential pattern $\alpha$ is called a *closed* pattern if there exists no sequential pattern $\beta$ such that $\alpha \sqsubset \beta$ and $Sup_a(\alpha) = Sup_a(\beta)$.

For example, in Table 2.4 patterns $< t_2, t_3 >$ and $< t_2 >$ appear three times in a document. However, pattern $< t_2 >$ is sub-sequence of pattern $< t_2, t_3 >$. Therefore, pattern $< t_2 >$ will be removed. Table 2.5 illustrates the set of closed sequential patterns in Table 2.4.

In the PTM, all closed sequential patterns can can be structured into a taxonomy by using the *is_a* (or *subset*) relation. Figure 2.6 illustrates the pattern taxonomy of closed sequential patterns in Table 2.5.

As shown in Figure 2.6, the pattern taxonomy is described as a set of pattern-absolute support pairs, for example $PT = \{< t_1, t_2, t_3 >_2, < t_2, t_3 >_3, < t_2, t_4 >_2$

| **Closed Sequential Patterns** | *Sup_a* | *Sup_r* |
|---|---|---|
| $< t_5, t_3 >, < t_1, t_2, t_3 >, < t_2, t_4 >$ | 2 | 0.5 |
| $< t_1 >, < t_2, t_3 >$ | 3 | 0.75 |
| $< t_3 >$ | 4 | 1.0 |

TABLE 2.5: All closed sequential patterns in the sample document in Table 2.3 with absolute support greater than or equal to 2

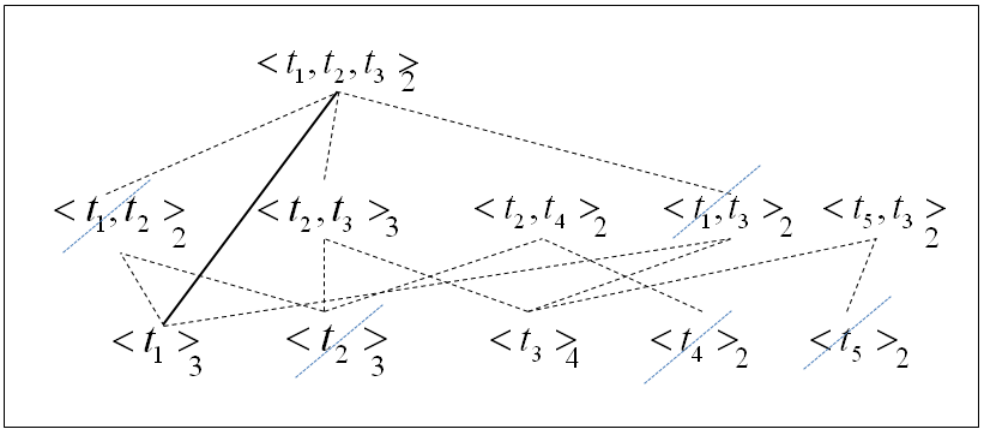



FIGURE 2.6: A pattern taxonomy of closed sequential patterns in the sample document in Table 2.4

}, where non-closed patterns are pruned. After pruning, some direct $is\_a$ relations may be changed, for example, pattern $< t_1 >_3$ would be come a direct sub-pattern of pattern $< t_1, t_2, t_3 >_2$ after pruning non-closed pattern $< t_1, t_2 >_2$.

#### 2.5.2.3 Basic Concept of Pattern Deploying

The deploying strategy proposed in [99] is mapping discovered patterns into a common hypothesis space to address the difficulties in using specific long patterns. Figure 2.7 illustrates the basic concept of pattern deploying.

As shown in Figure 2.7, high-level patterns in the pattern space represent collections of low-level terms in the term space (or a topic). Such collections have

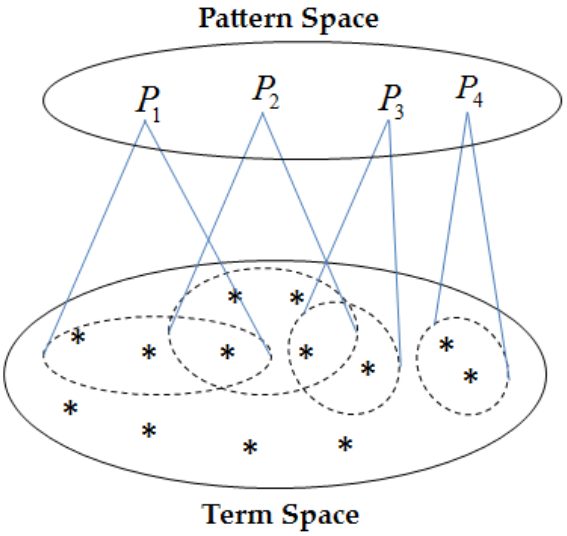



FIGURE 2.7: The Concept of Pattern Deploying

usually overlapping terms which reflect their importance in the underlying topic due to they are frequently refereed. Based on this assumption, the deploying method evaluates a given term's support based on its appearances in the patterns.

the deploying method is described as follows. Let $SP_1, SP_2, \ldots, SP_{|D^+|}$ be the sets of closed sequential patterns for each relevant document $d_i \in D^+$, where $i = 1, \ldots, |D^+|$. Let $T$ be a set of terms in the documents. For each term $t \in D^+$, the term support can be calculated by:

$$support(t, D^+) = \sum_{i=1}^{n} \sum_{t \in p \subseteq SP_i} \frac{sup(p, d_i)}{|p|} \tag{2.13}$$

where $|p|$ is the number of terms in $p$.

Table 2.6 illustrates an example of sets of discovered closed sequential patterns

| **Doc.** | **Discovered Closed Sequential Patterns ($SP_i$)** |
|---|---|
| $d_1$ | $PT_1 = \{< t_1 >_4, < t_1, t_2 >_3, < t_3, t_4 >_2\}$ |
| $d_2$ | $PT_2 = \{< t_5, t_6 >_3, < t_2, t_6 >_2\}$ |
| $d_3$ | $PT_3 = \{< t_5 >_2, < t_6, t_2 >_2\}$ |
| $d_4$ | $PT_4 = \{< t_1 >_3, < t_3 >_3, < t_3, t_7 >_2\}$ |
| $d_5$ | $PT_5 = \{< t_2, t_6, t_8 >_2\}$ |

TABLE 2.6: A set of documents and their pattern taxonomy

for $D^+ = \{d_1, d_2, d_3, d_4, d_5\}$. For example, team $t_6$ appears in three documents, i.e., $d_2$, $d_3$, and $d_5$. Based on this weighting, its support is evaluated based on patterns the sets of closed sequential patterns that contain $t_6$, i.e.,

$$support(t_6) = w(t_6, d_2) + w(t_6, d_3) + w(t_6, d_5)$$

$$support(t_6) = (\frac{3}{2} + \frac{2}{2}) + (\frac{2}{2}) + (\frac{2}{3}) = 2.5 + 1 + 0.667 = 4.16$$

Moreover, the support of term $t_4$ in the training documents can be calculated, i.e.,

$$support(t_4) = w(t_4, d_1) = \frac{2}{2} = 1$$

and the support of term $t_5$ is

$$support(t_5) = w(t_5, d_2) + w(t_5, d_3) = (\frac{3}{2}) + (\frac{2}{1}) = 1.5 + 2 = 3.5.$$

Once the all term supports are evaluated, a document evaluation is formed to use the extracted low-level features for scoring a test document $d$ as follows:

$$rank(d) = \sum_{t \in T} support(t, D^+)\tau(t, d) \tag{2.14}$$

where $T$ be a set of extracted features; and $\tau(t, d) = 1$ if $t \in d$; otherwise $\tau(t, d) = 0$.

### 2.5.3 Negative Relevance Feedback

Some previous studies show that there is plenty of useful non-relevant information available in negative relevance feedback [6, 93, 112]. However, effectively using negative relevance feedback to improve largely filtering accuracy is still an open issue [60].

The existing methods for using negative feedback in IF have been largely proposed in traditional IR and IF. However, little work has been done in pattern-based approaches. The traditional IR-based and IF-based models have been done by using machine learning algorithms. For example, Rocchio-based models [46] and SVM-based filtering models [25]). By using these machine learning techniques, the problem of extracting relevant features for IF can be treated as the problem of binary classification. All terms extracted from positive training samples can be used as features to distinguish the difference between relevant and non-relevant documents.

Recently, a pattern-based model for using negative relevance feedback in PTM has been proposed [112]. This approach uses information in non-relevant documents to improve the quality of extracted features. The result shows that using

negative relevance feedback has better effectiveness results. However, the improvement gain of the model that uses both feedback is not significant compared to the model using only relevance feedback. The key challenge is how to utilise negative relevance feedback for pattern-based approaches to improve the quality of extracted features in user relevance feedback.

## 2.6 Chapter Summary

In this chapter, the overview of knowledge discovery process as well as frequent pattern mining has been reviewed. We also review the current work in pattern mining techniques for finding useful patterns in databases, which try to overcome the quality issue of using frequent pattern mining.

In terms of IR and IF, we review a lot of work done in the particular area, varying from classical approaches in IR to current approaches in data mining. Finally discussion of these RFD approaches were introduced.

# Chapter 3

# The Pattern Mining Framework

As mentioned in 2, pattern taxonomy models (PTM) that utilise closed sequential patterns in text documents to overcome the limitation of traditional term-based approaches. However, the key challenge of PTM is how to effectively deal with numerous discovered patterns for the extraction of accurate features. Among discovered patterns, there are many meaningless patterns, and also some discovered patterns may include general information (i.e., terms or phrases) about the user's topic. Such patterns are noisy and often restrict effectiveness [112].

This chapter presents a novel data mining framework for acquiring user information needs or preferences in text documents. This framework utilises pattern taxonomy mining [98] to capture important semantics information in a feedback set of relevant documents. After that, a new post-mining method, named *pattern cleaning*, for relevance feature discovery, is applied to reduce the effects of noisy information captured by pattern mining. Finally, relevance feature models for using the knowledge patterns are employed to help users search for accurate

information. Figure 3.1 illustrates the pattern mining framework for relevance feature discovery.

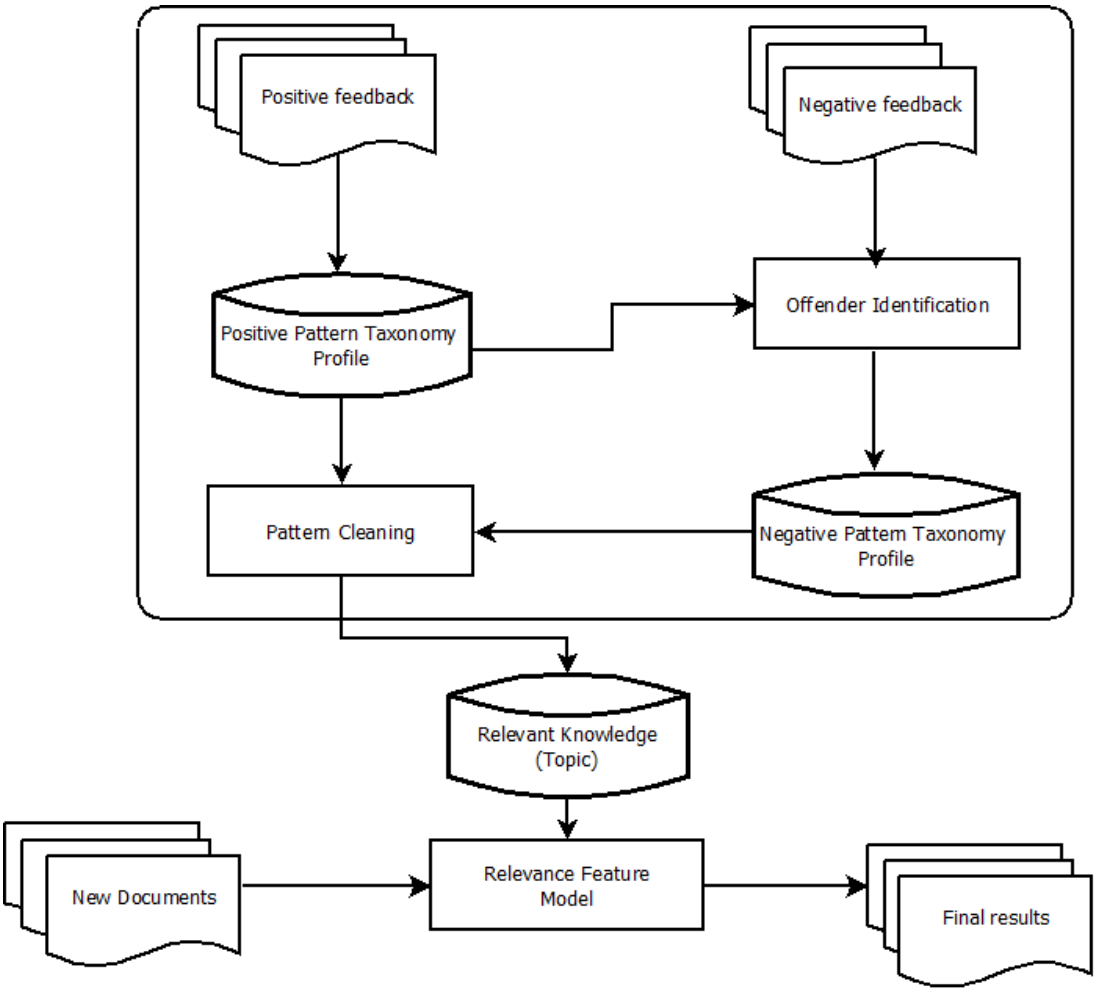



FIGURE 3.1: The proposed framework

As shown in Figure 3.1, our proposed framework consists of three major parts: (1) pattern taxonomy mining, (2) pattern cleaning, and (3) relevance feature model. In the following sections, we briefly explain each part of the proposed framework.

## 3.1 Pattern Taxonomy Mining

As mentioned in 2, pattern taxonomy provides a nice way to view possible document representation methods. The new representation captures more important semantics information with closed sequential patterns in text documents and goes beyond the classical term-based representation. Closed sequential patterns have been shown to be useful as phrases for information retrieval (IR) and text mining [29, 71, 99, 112]. The rationale behind the frequent phrases is that such phrases

- capture the semantic relationships among words in text documents and allow a gap between the words, offering rich semantic information about natural language [24];
- allow researchers to work with a variety of text features, including single words, short phrases and long phrases;
- provide concise lossless representation of the original set of frequent patterns; and remove noisy short patterns with respect to information redundancy [112].

In this thesis, the concept of pattern taxonomy is employed for document and topic representation in a feedback set.

Given a feedback set of relevant (positive) documents and non-relevant (negative) documents, taxonomy profiles of closed sequential patterns and their relations in the feedback set are extracted. Such profiles contain meaningful relevant and non-relevant information for topic representation.

Nevertheless, the extraction of pattern taxonomy from large document collections can be infeasible because of the large number of discovered patterns. This research presents an efficient algorithm for pattern taxonomy mining, called *PTMining* in order to speed up the mining process. *PTMining* algorithm will be explained in Chapter 4.

## 3.2 Pattern Cleaning

For a given topic, closed sequential patterns extracted from positive (relevant) documents (or *positive patterns* for short) capture prices of meaningful information for describing user information needs. However, a lot of meaningless and irrelevant information available in the feedback documents can easily affect the quality of extracted features. Using closed pattern mining cannot deal with the noisy information subject to a specific need of user. For example, short patterns with high support generally contain general information for a specified topic, but specific long patterns have low support [112].

The objective of pattern cleaning is to reduce the effects of noises caused by the discovery process. The main idea of pattern cleaning is to utilise non-relevant information to refine the relevant knowledge for a specified topic. However, using all negative documents may be not interesting and increase noises since they may be often collected from other topics. In this research, we introduce the notion of *offenders* to address the above issue. An offender is defined as a negative document that is closer to positive ones. According to Figure 3.1, the set of offenders are identified from a feedback set of negative documents by using some positive

patterns. After that, the offenders are employed to refine positive patterns for accurate relevant knowledge (i.e., topics). The pattern cleaning and offender selection will be explained in Chapter 5.

## 3.3 Relevance Feature Model

Once the relevant knowledge is extracted from a feedback set, the next step is how to utilise the discovered knowledge for the effectiveness of retrieval system. In this thesis, information filtering (IF), an application in IR and text mining, will be studied to evaluate the quality of relevant knowledge for a user's topic.

In order to utilise the relevant knowledge, this research presents two representation models for the relevant knowledge. The first model is to treat patterns as high-level features to find accurate information of user. A new feature weighting method is applied to assign accurate weights to each pattern in order to reflect their significance in the user's topic. Finally, the relevance of an incoming document is evaluated based on the appearances of the patterns in the document.

The second representation model for using the relevant knowledge is aimed to address the problem of using specific long patterns in text. We develop equations to deploy high-level patterns over low-level features (term) using term support based on their appearance in patterns. Finally, the set of low-level features is employed to improve filtering performance instead of original patterns.

## 3.4 Chapter Summary

In this chapter, we present a novel framework of pattern mining for relevant feature discovery. This framework integrates data mining methods to enhance both efficiency and effectiveness of feature discovery in positive and negative feedback for describing user's information needs.

Basically, these data mining methods include (1) pattern taxonomy mining, (2) pattern cleaning, and (3) relevance feature model. The pattern taxonomy mining will be described in Chapter 4. The pattern cleaning method will be explained in Chapter 5. Finally, the relevance feature models for using the discovered knowledge in a feedback set will be described in Chapter 6.

# Chapter 4

# Pattern Taxonomy Mining

## 4.1 Introduction

As discussed in Chapter 3, the desirable properties of closed sequential patterns can be potentially useful for knowledge discovery in text. However, the discovery of the closed patterns typically produces a large collection of patterns, which may hinder their use or the extraction of useful patterns.

In this chapter, we propose a novel index structure which allows to deal with a large collection of closed sequential patterns. We also develop a direct mining technique, called *Pattern Taxonomy Mining (PTMining)*, for building this index during the mining process. In the following sections, we review the basic definitions of sequential patterns and closed sequential patterns. Then, we define the problem of closed sequential patterns in text and its algorithm. Finally, PTMining algorithm will be described in this chapter.

## 4.2 Pattern Taxonomy Mining

Although closed sequential patterns can result in significantly reducing the number of sequential patterns, the reduction remains a large collection and may include noisy patterns. In many cases, the huge amount of closed patterns may hinder to find useful knowledge for the user feedback due to the large search space.

In this section, we describe a novel indexing mechanism for closed sequential patterns, called *pattern taxonomy*. A pattern taxonomy is a tree-like structure that contains a collection of closed sequential patterns and their relation in a feedback set of documents. The main objective of the use of pattern taxonomy is to prepare numerous discovered closed patterns in the feedback set for efficient processing of pattern cleaning. In this thesis, a data mining algorithm, called *PTMining*, has proposed for the extraction of pattern taxonomy in a feedback set.

Let us consider a sample document collection in Table 4.4.

| **Document** | **Paragraph-id** | **Term Sequence** |
|---|---|---|
| $d_1$ | $dp_{11}$ | $t_1\ t_2\ t_3\ t_4$ |
| | $dp_{12}$ | $t_1\ t_6\ t_3\ t_7$ |
| | $dp_{13}$ | $t_6\ t_7\ t_8$ |
| | $dp_{14}$ | $t_6\ t_7$ |
| $d_2$ | $dp_{21}$ | $t_{10}\ t_6\ t_7$ |
| | $dp_{22}$ | $t_6\ t_9\ t_7\ t_8$ |
| | $dp_{23}$ | $t_7\ t_4$ |

TABLE 4.1: A sample document collection

According to Table 4.4, this text collection has documents $d_1$ and $d_2$, where $d_1$ and $d_2$ consist of four and three paragraphs respectively. As mentioned in

the previous section, PTM model [98] produces collections of closed sequential patterns for the documents. However, the collections of patterns often represent fragmented knowledge and often it is not clear how these collections can be combined to obtain a global view of the patterns.

For example, given the document collection in Table 4.4 and $min\_sup = 2$, sequential pattern $< t_1, t_3 >$ can be reported in document $d_1$, but not in document $d_2$. On the other hand, sequential pattern $< t_6, t_7 >$ that contains in the both documents has different supports. The absolute support of pattern $< t_6, t_7 >$ in document $d_1$ is 3 (or 0.75 for $Sup_r$) while its support in document $d_2$ is 2 (or 0.66 for $Sup_r$).

In PTMining, the documents are transformed into a sequence database by assuming that each paragraph of these documents is a transaction of the database. Table 4.5 illustrates the sequence database of the documents in Table 4.4.

| **Paragraph ID** | **Term Sequence** |
|---|---|
| $dp_{11}$ | $t_1\ t_2\ t_3\ t_4$ |
| $dp_{12}$ | $t_1\ t_6\ t_3\ t_7$ |
| $dp_{13}$ | $t_6\ t_7\ t_8$ |
| $dp_{14}$ | $t_6\ t_7$ |
| $dp_{21}$ | $t_{10}\ t_6\ t_7$ |
| $dp_{22}$ | $t_6\ t_9\ t_7\ t_8$ |
| $dp_{23}$ | $t_7\ t_4$ |

TABLE 4.2: A sequence database of the documents in Table 4.4

According to Table 4.5, the sequence database consists of seven sequences of terms in all paragraphs of document $d_1$ and $d_2$ in Table 4.4. Duplicate sequences are also removed to reduce the size of database.

Once the database was built, PTMining algorithm can be applied to construct a profile that contains all closed sequential patterns and their relation discovered in the database. However, a major problem is that the size of the database can be quite large, which may pose more challenges on the mining efficiency. Thus, efficient mechanisms are used to reduce the search space of sequential patterns. In the following subsections, we describe the mechanisms used in PTMining algorithm for the efficient mining process. Then, we present this algorithm in the last section.

### 4.2.1 Pattern Taxonomy Profile

The output of PTMining is a taxonomy profile that contains closed sequential patterns and their "is-a" (sub-pattern/ super-pattern) relation in a sequence database. Figure 4.1 illustrates the taxonomy structure of discovered patterns with support (i.e., the subscripts of <>) in the database shown in Table 4.5.

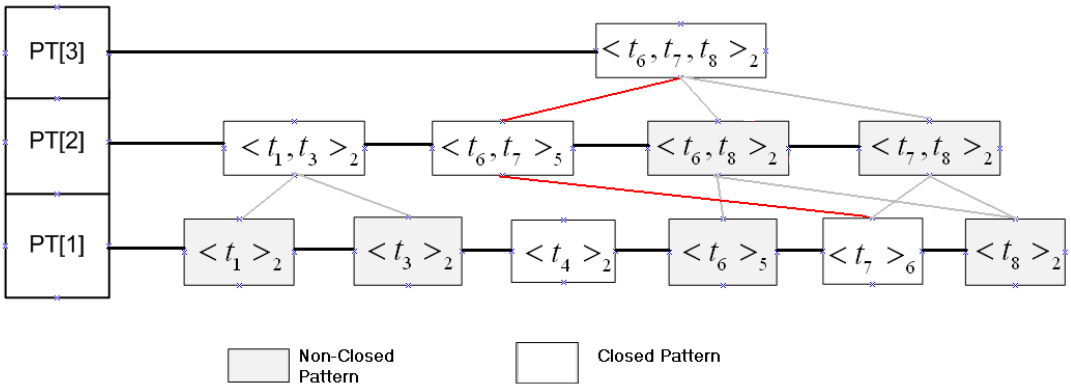



FIGURE 4.1: The taxonomy profile for sequential patterns in Table 4.5 at $min\_sup = 2$

According to Figure 4.1, the taxonomy profile contains three lists of patterns in the database shown in Table 4.5 according to their length. For example, $PT[1]$ contains the list of all frequent terms (or called size-1 patterns) which include $t_1$, $t_3$, $t_4$, $t_6$, $t_7$, and $t_8$ respectively. $PT[2]$ and $PT[3]$ contain the lists of size-2 and size-3 patterns respectively.

The edges that connects the patterns in different lists represent their relation. For example, pattern $< t_1 >$ in list $PT[1]$ has a direct edge to pattern $< t_1, t_3 >$ in list $PT[2]$ since $< t_1 >$ is a sub-pattern of $< t_1, t3 >$. Pattern $< t_4 >$ in list $PT[1]$ has no edge to the others because there is no its super-pattern found in the sequence database.

To obtain the close set of patterns, non-closed patterns are removed from the profile (i.e., shared rectangles) and the edges needs to be maintained. For example, patterns $< t_1 >$ and $< t_3 >$ in list $PT[1]$ are eliminated from this profile since their support is the same as pattern $< t_1, t_3 >$ which is super-pattern of the patterns. Furthermore, pattern $< t_7 >$ has only a direct edge to pattern $< t_6, t_7 >$ in list $PT[2]$ since pattern $< t_7, t_8 >$ is non-closed pattern which is removed from this profile.

### 4.2.2 Sequence Extension

The project-database technique have been proposed for efficient sequential pattern mining on large databases [37]. The main idea of this technique is to avoid generating candidates for sequence extension of a pattern by partitioning an original database into a smaller set of extended sequences of the pattern. Given a pattern $\alpha$, the formal definition of $\alpha-$projected database can be given as follows:

*Definition* 4 ($\alpha-$**projected database**). Let $\alpha$ be a sequential pattern in sequence database $S$. The $\alpha-$projected database, denoted as $S_\alpha$, is the set of postfixes of sequences in $S$ which are made of prefix $\alpha$.

*Definition* 5 (**Support in projected database**). Let $\alpha$ be a sequential pattern in sequence database $S$ and $\beta$ be a sequence having prefix $\alpha$. The support of $\beta$ in $\alpha-$projected database $S_\alpha$ is number of sequences $\gamma$ in $S_\alpha$ such that $\beta \sqsubseteq \alpha \bowtie \gamma$.

For example, referred to the sample database in Table 4.5, let $\alpha =< t_7 >$ be a sequential pattern in the database. The $\alpha-$projected database contains a collection of sequences w.r.t. prefix $\alpha$, which include $dp_{12} :<>$, $dp_{13} :< t_8 >$, $dp_{14} :<>$, $dp_{21} :<>$, $dp_{22} :< t_8 >$, and $dp_{23} :< t_4 >$ where $<>$ is *null*. Let $\beta =< t_7, t_8 >$ be an extended sequence of $\alpha$. The support of $\beta$ in the $\alpha-$projected database is 2.

Once the $\alpha-$projected database for pattern $\alpha$ was extracted, the next step is to find the sequential patterns in this database satisfying a given minimum support. If one frequent term is found, a $(n+1)$ sequential pattern is expanded from pattern $\alpha$ by using sequence extension, which is defined as follows:

*Definition* 6 (**Sequence Extension**). Given $\beta$ be a sequence in a database and a sequential pattern $\alpha$, the sequence extension $\gamma$ of $\alpha$ w.r.t. $\beta$ is obtained by simply appending $\beta$ to $\alpha$ and generating a new sequence $\gamma$ such that $\gamma = \alpha \bowtie \beta$.

Here, we illustrate the process of sequence extension in the sequence database shown in Table 4.5. Given $min\_sup = 2$, the list of size-1 patterns generated from the original database can be seen in Table 4.6.

| **size-1 pattern** | $Sup_a$ | $Sup_r$ |
|---|---|---|
| $< t_1 >$ | 2 | 0.28 |
| $< t_3 >$ | 2 | 0.28 |
| $< t_4 >$ | 2 | 0.28 |
| $< t_6 >$ | 5 | 0.62 |
| $< t_7 >$ | 6 | 0.85 |
| $< t_8 >$ | 2 | 0.28 |

TABLE 4.3: Size−1 frequent patterns for the database in Table 4.5 where $min\_sup = 2$

For each size-1 pattern $\alpha$, a set of subsequences $P_s$ starting with $\alpha$ are extracted from the original database, where $P_s \sqsubseteq dp_n$ and $dp_n \in d$. Generally, the number of projected sequences of a root pattern $\alpha$ equals to the pattern's absolute support unless it locates at the end of some paragraphs.

After a $\alpha-$projected database is built, size-$(n+1)$ patterns can be obtained by extending the size$-n$ pattern using the concept of sequence extension. For example, in Table 4.7 extended sequences $< t_2, t_3, t_4 >$ and $< t_6, t_3, t_7 >$ are generated with a root pattern $< t_1 >$. Then, only a candidate $< t_1, t_3 >$ with the support of 2 is generated because term $t_3$ is frequent in the projected database. Table 4.7 illustrates the $\alpha-$ projected database for all size-1 patterns in Table 4.6.

Table 4.8 illustrates the results of size-2 patterns derived from the projected databases in Table 4.7. Note that neither pattern will be generated from the $p-$projected database of pattern $< t_4 >$ and $< t_8 >$ because there is no extended sequence existing in their projected database. Furthermore, there is no pattern will be generated from the $\alpha-$projected database of pattern $< t_3 >$ because terms $t_4$ and $< t_5 >$ are not frequent in its projected database.

| $\alpha$ | Extended Subsequence |
|---|---|
| $< t_1 >$ | $< t_2, t_3, t_4 >$<br>$< t_6, t_3, t_7 >$ |
| $< t_3 >$ | $< t_4 >$<br>$< t_7 >$ |
| $< t_4 >$ | |
| $< t_6 >$ | $< t_3, t_7 >$<br>$< t_7, t_8 >$<br>$< t_7 >$<br>$< t_7 >$<br>$< t_9, t_7, t_8 >$ |
| $< t_7 >$ | $< t_8 >$<br>$< t_8 >$<br>$< t_4 >$ |
| $< t_8 >$ | |

TABLE 4.4: The list of $\alpha-$projected databases for all size-1 patterns in Table 4.6

| **size-1 pattern** | **size-2 pattern** | $Sup_a$ | $Sup_r$ |
|---|---|---|---|
| $< t_1 >$ | $< t_1, t_3 >$ | 2 | 0.28 |
| $< t_3 >$ | – | – | – |
| $< t_4 >$ | – | – | – |
| $< t_6 >$ | $< t_6, t_7 >$ | 5 | 0.62 |
| | $< t_6, t_8 >$ | 2 | 0.28 |
| $< t_7 >$ | $< t_7, t_8 >$ | 2 | 0.28 |
| $< t_8 >$ | – | – | – |

TABLE 4.5: The list of size-2 patterns derived from all size-1 patterns

This process continues to extend sequential patterns of size$-n$ until there is no more extended sequences.

### 4.2.3 Efficient Closure Checking

When a new sequential pattern is generated, closure checking needs to be applied for eliminating non-closed sequences. Let $\alpha$ be a new sequential pattern in a sequence database.

The problem of closure checking is to check out for each closed sequence $\beta$, whether there exists a super-sequence $\alpha$ and the support of $\beta$ equals to the support of $\alpha$. The bottleneck of closure testing is to compare each closed sequence already mined, which gives $O(n^2)$, where $n$ is the total number of sequential patterns discovered in the database.

To overcome this difficulty, some efficient algorithms for closed pattern mining have been proposed, such as CLOSET [70], CHARM [110], and CLOSET+ [92], CloSpan [105]. Most of them proposed for efficient mining of closed itemsets, which may be not suitable for subsequence closure checking due to ordered matching required.

PTMining algorithm adopted the *candidate-maintenance-and-test* approach used in CloSpan algorithm [105] for efficient elimination of non-closed sequences in a pattern profile. Basically, a hash-indexed structure is used to maintain a set of closed sequence candidates already mined and then do post-pruning. To implement the index structure, PTMining uses the open hash table which allows to store a set of closed candidates. Figure 4.2 illustrates the hash-indexed structure used in PTMining.

According to Figure 4.2, the hash structure consists of lists of objects (or open chaining) stored in different locations of the hash table. The hash function $h()$

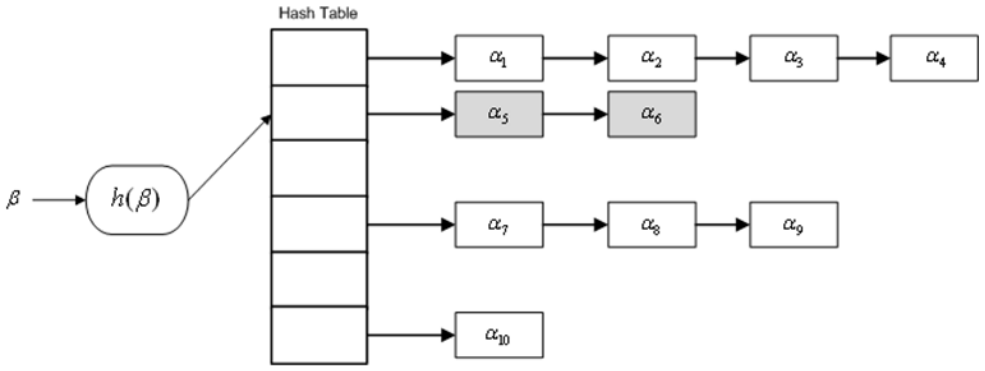



FIGURE 4.2: The Hash-Indexed Structure

is used to compute a hash key for matching a new object $\beta$ to a specific location in the hash table that stores objects with the same key. For example, the hash function in Figure 4.2 matches object $\beta$ to a list of objects $\alpha_5$ and $\alpha_6$ in the hash table.

However, the objective of using the hash structure is not to index new patterns, but is to reduce the search space of subsequence testing. Assume that a set of closed candidates are stored in the hash table. When a new sequential pattern is generated by the sequence extension, a hash key of the new pattern is computed by using a hash function. Finally, a subset of the closed candidates in the hash table is retrieved to perform closed subsequence checking based on the common key.

Since closure checking uses support information to compare sequences, we define to use the support of pattern for calculating a hash key in the hash function. Let $h(Sup_a(\alpha))$ be a hash function that assigns a key value to pattern $\alpha$ based on its support. Once the key is calculated, the hash index $HT$ retrieves a set of closed

candidates $\beta$ based on their common key, i.e.,

$$C_\alpha = \{\beta | \beta \in HT, h(Sup_a(\beta)) = h(Sup_a(\alpha))\} \tag{4.1}$$

By using the hash index, we can reduce the search space of closed sequence checking by testing only a subset of them. Once the closed candidates of pattern $\alpha$ was identified, we do post-pruning, where a closed candidate is removed from the profile if there exists a super-pattern $\alpha$.

Figure 4.3 illustrates the example of the use of the hash index for non-closed sequence elimination. Assume that the hash index now stores sequential pattern-ids ($Seq.ID$) of size-1 patterns in list $PT[1]$ of the taxonomy according to their key. For example, patterns $t_1$ and $t_2$ are stored in the same location because they have the same support in the sequence database, i.e., 3. When a new pattern $p =< t_2, t_3 >$ with support of 3 is extracted, a hash key of the pattern is calculated by the hash function. Then, the hash index returns a set of size-1 candidates with the common key for closure checking. For example, applying the hash key for pattern $< t_2, t_3 >$ results in testing only closed patterns $t_1$ and $t_2$ because they have the common key. A closed candidate that is subsequence of the new pattern is eliminated from the pattern profile; otherwise, it is still closed one.

By this way, we do not need to compare patterns $t_4$, $t_3$, and $t_5$ which have a different key from $< t_2, t_3 >$. As a result, the search space of closed candidates can be largely reduced. We do the same with the remaining new frequent patterns.

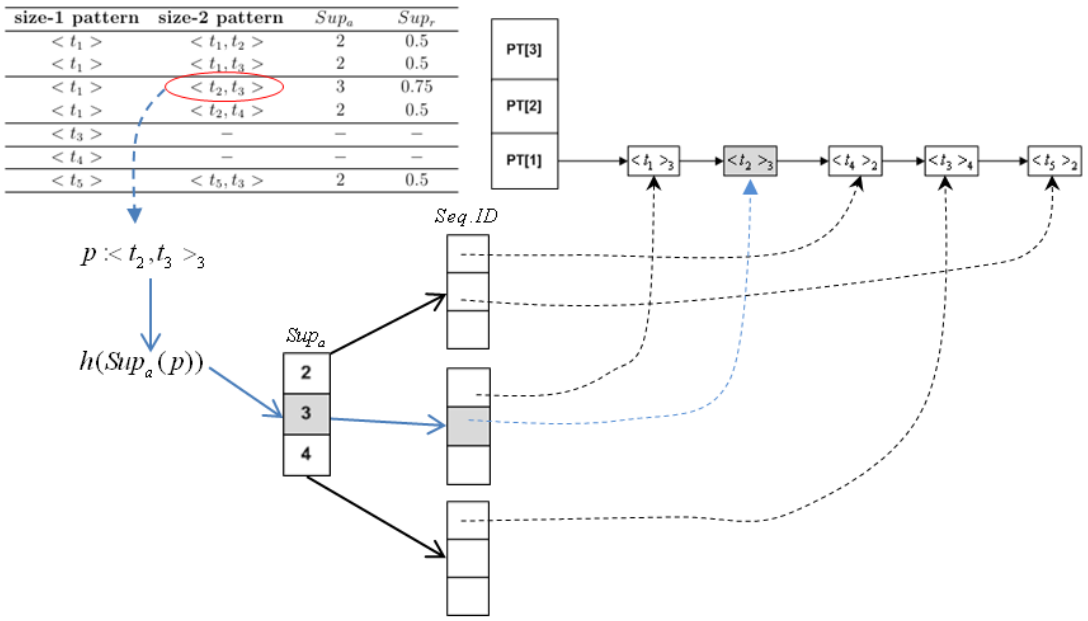



FIGURE 4.3: The pattern taxonomy for frequent patterns and closed patterns in Table 4.5

## 4.3 Algorithm Implementation

Algorithm 1 describes PTMining algorithm. The input of this algorithm includes a set of feedback documents, denoted as $D$, and a minimum support threshold $min\_sup$. The output of this algorithm is a taxonomy profile $PT_D$ that contains closed sequential patterns and their relation in the document set.

PTMining algorithm starts by partitioning each document of the feedback collection into a set of paragraphs to build a sequence database $S_D$ (Step 2 to 5). It uses the sequence database to mine closed sequential patterns for the feedback documents. After that, a set of all frequent terms (or 1 term sequences) is extracted from the sequence database $S_D$ to initialize the pattern profile for mining sequence extension by inserting them into list $PT_D[1]$ (Step 7 to 10). Step

11 calls *ClosedSeqMine* algorithm for mining closed sequential patterns. The ClosedSeqMine can be seen in Algorithm 2.

**Algorithm 1**: PTMining

**Input** : A feedback collection $D$; a minimum support $min\_sup$
**Output**: a taxonomy profile $PT_D$;
**begin**

```
1   S_D = ∅;
2   for each document d ∈ D do
3       Let PS(d) be a set of paragraphs in d;
4       S_D = S_D ∪ PS(d) ;                    /* Build a Sequence Database */
5   Remove duplicate sequences in S_D;
6   PT_D[1] = ∅;
7   Let T be a set of all frequent 1-term sequences in S_D;
8   for each term t ∈ T do
9       t.child = NULL;
10      PT_D[1] = PT_D[1] ∪ {t} ;              /* Insert Frequent Terms */
11  Call ClosedSeqMine(PT_D[1], S_D);
12  return
```

**end**

In *PTMining* procedure, Level $n+1$ ($PT^+[n+1]$) is initialized in step 7, and candidate patterns of size $(n+1)$ are iteratively generated by joining all their sub-patterns already mined in the current level, where $n$ means the current level (See Steps 8 to 11). Steps 12 to 14 examine all the new candidates and only the ones that have a frequency above the minimum threshold $min\_sup$ are inserted into $PT^+[n+1]$ as new frequent patterns discovered. The process of updating the taxonomy when removing non-closed patterns is described from Steps 15 to 20. For each frequent pattern $\beta$ in $PT^+[n+1]$, its all sub-patterns are extracted from the list $PT^+[n]$ (i.e., $\sigma_\beta$) in Step 15. For each sub-pattern $q \in \sigma_\beta$, pattern

**Algorithm 2**: ClosedSeqMine

**Input** : The list of size$-n$ closed sequential patterns: $PT_D[n]$; A sequence database $D$; A hash-index: $H$

**Output**: The list of size$-(n+1)$ closed sequential patterns: $PT_D[n+1]$;

**begin**

```
1  Let PT_D[n+1] = ∅;
2  for each pattern α ∈ PT_D[n] do
3      Insert α to H with h(Sup_a(α));
4  for each pattern α ∈ PT_D[n] do
5      Generate α−projected database S_{D|α} ;
6      for each term t ∈ S_{D|α} do
7          γ ← α ⋈ t ;                          /* Sequence Extension */
8          if Sup_a(γ) ≥ min_sup then
9              γ.child = NULL;
10             PT_D[n+1] = PT_D[n+1] ∪ {γ} ; /* Add a new Seq Pattern */
11             C_γ = {β|β ∈ H, h(Sup_a(β)) = h(Sup_a(γ))};
12             for each candidate β ∈ C_γ do
13                 if β ⊑ γ then
14                     γ.child = β.child;
15                     PT_D[n] = PT_D[n] − {β};
16                     else γ.child = β ;  /* Prune Non-Closed Sub-Ptrns */
17 if PT_D[n+1] = ∅ then return ;     /* No More Frequent Patterns */
18 Call ClosedSeqMine(PT_D[n+1],S_D);
```

**end**

$q$ is removed from the $PT^{+}[n]$ if its support is equal to the support of pattern $\beta$; otherwise, pattern $q$ is added as a child of pattern $\beta$ (Step 20). Finally, the *PTMining* algorithm keeps expanding the tree till no frequent patterns of size $(n+1)$ are generated (Step 20).

# Chapter 5

# Pattern Cleaning

The main challenge that limits the practical usage of pattern mining is that typical data mining algorithms generate patterns in numbers too large to be useful [39]. Among discovered patterns, there are many meaningless patterns, and also some discovered patterns might include uncertain information as well. Over the years, many data mining techniques have been proposed for removing redundant and noisy patterns in data. For text classification, pattern taxonomy models [98, 99, 112] that have adopted the concept of closed patterns to remove meaningless patterns in text with respect to others have shown a certain improvement on text classification performance. However, we believe that the closed frequent patterns are insufficient to address the problem of noisy patterns in relevant documents. The first reason is that closed pattern mining focuses on reducing a large collection of frequent patterns in a single set of data. As a result, the closed patterns suffer from from large amounts of general patterns for describing a specified topic. The second reason suggested by [112] is the problem

of misinterpretation, which means that highly frequent patterns (normally short patterns with large support) are usually general patterns, but long patterns with low support (normally specific patterns).

This chapter describes a novel post-mining method for reducing noises in closed sequential patterns, which lead to improve the quality of extracted features in user relevance feedback.

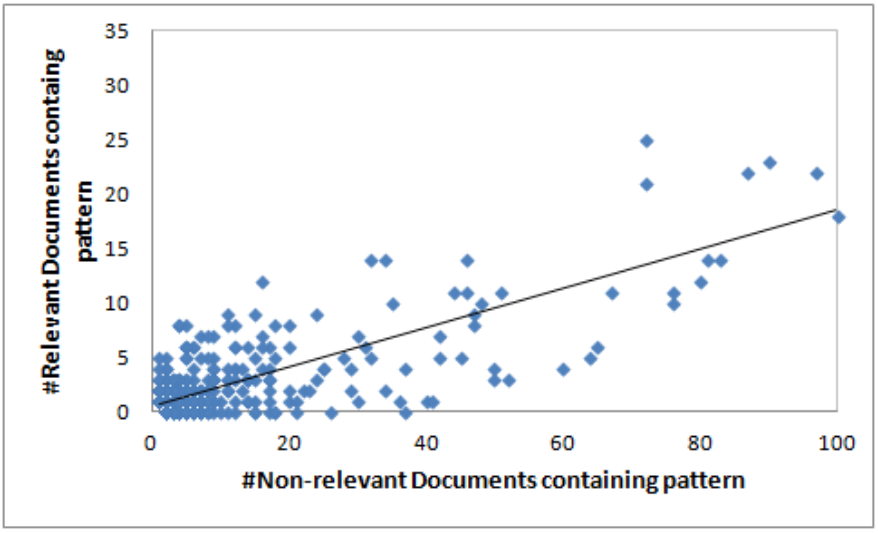
## 5.1 Presence of Noise in Relevance Feedback

User feedback contains useful information about how the users personally satisfy the results of objects (typically documents) retrieved by a system [78]. A typical collection of user feedback includes *relevant* and *non-relevant* documents with respect to user's perspectives. For data mining, it is expected that the discovery of patterns in user relevance feedback should capture useful and semantic information for describing user information needs or preferences.

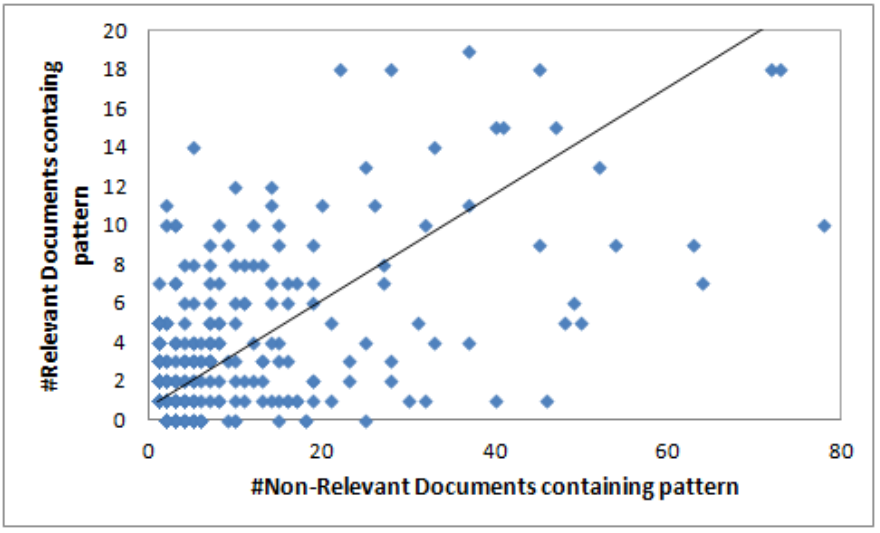
Real-world text collections in general contain a lot of meaningless and uncertain information (i.e., terms or phrases). For example, large documents may contain many terms that span several subject areas. When the documents are employed for the purpose of relevance feedback, large amounts of noisy terms can be extracted by using pattern mining algorithms. Such noisy patterns can not only increase time complexity, but also increase noise in discovered patterns, which finally restrict effectiveness [104].

Let us show the problem of mining frequent patterns in relevant documents that may contain noise. Figure 5.1 illustrates the frequency counts (or support) of

close sequential patterns in relevant and non-relevant documents of some feedback sets in RCV1 corpus [55]. The blue dots represent the true positive and false positive counts for the frequent patterns from relevant documents of the feedback sets.




(a)




(b)

FIGURE 5.1: Frequent pattern mining with RCV1's topics

As shown in Figure 5.1, large amounts of closed patterns in the training datasets are *general patterns*, which appear in both relevant and non-relevant documents of a feedback set. Furthermore, such patterns tend to be highly frequent patterns in the datasets. This can imply that relevant documents share common information with non-relevant ones. Given a specified topic, a general pattern usually has a *high exhaustivity* but a *low specificity*, where exhaustivity describes the extent to which a pattern discusses the topic and specificity describes the extent to which a pattern focuses on this topic.

According to the noise characteristic, let us define the term "noise" used throughout the thesis. Let $A$ be a set of frequent patterns in a feedback set $D$, and $IF_A$ be an information filtering system implemented by using $A$. Let $P$ be a positive pattern discovered in $D$, $P \notin A$ and $IF_{AP}$ be the information filtering system implemented by using $A \cup \{p\}$. We called pattern $p$ a *noise* pattern to $A$ if the performance of $IF_{AP}$ is worse than $IF_A$.

Ideally, it is expected to remove the large amounts of noisy pattern $P$ in discovered patterns to improve the performance of information filtering systems. For many years, data mining has developed many post-mining techniques for removing meaningless and noisy patterns. Most of these methods focus on summarizing the large collection of frequent patterns using fewer patterns in an unsupervised setting. However, without considering non-relevant information it is impossible to deal with the noises in relevant documents. Recently, some pattern mining techniques for finding patterns that are relevant to the class of interest, called relevant patterns, have been proposed and studied under the names of emerging patterns [22, 56] and discriminative patterns [17, 27]. The main idea of these

methods is to mine a complete set of frequent patterns and then select relevant patterns based on some criteria. However, the key challenge is how to correctly decide which patterns should be removed because the boundary between relevant and noisy patterns for a specified topic is not clear.

## 5.2 Noise Reduction Approach

In this section, we propose the idea of pattern cleaning to improve the quality of relevant knowledge extracted in noisy feedback. Basically, the proposed method mines both relevant and non-relevant information.

It is clear that non-relevant documents contain information not desired by the user. This feedback information are useful to identify ambiguous patterns with respect to user's information needs. For example, a pattern is ambiguous if it appears in both the relevant and non-relevant documents at certain times. Furthermore, as the very limited number of relevant documents in a typical feedback collection, the pieces of relevant information extracted from the training documents are hardly representative for describing the relevant knowledge. By combining non-relevant information, the quality of relevant knowledge can be expected due to additional information.

## 5.3 Pattern Cleaning Method

As discussed in the previous section, the challenging problem of mining relevance feedback is that it generates a lot of detailed information (patterns), which may

be not good representative for describing user feedback and may also include some noisy patterns. The main idea of pattern cleaning is to try updating the relevant information by the use of non-relevant information to improve the quality of discovered knowledge. Figure 5.2 illustrates the proposed method of pattern cleaning. It consists of two main steps: (1) offender identification and (2) pattern refinement.

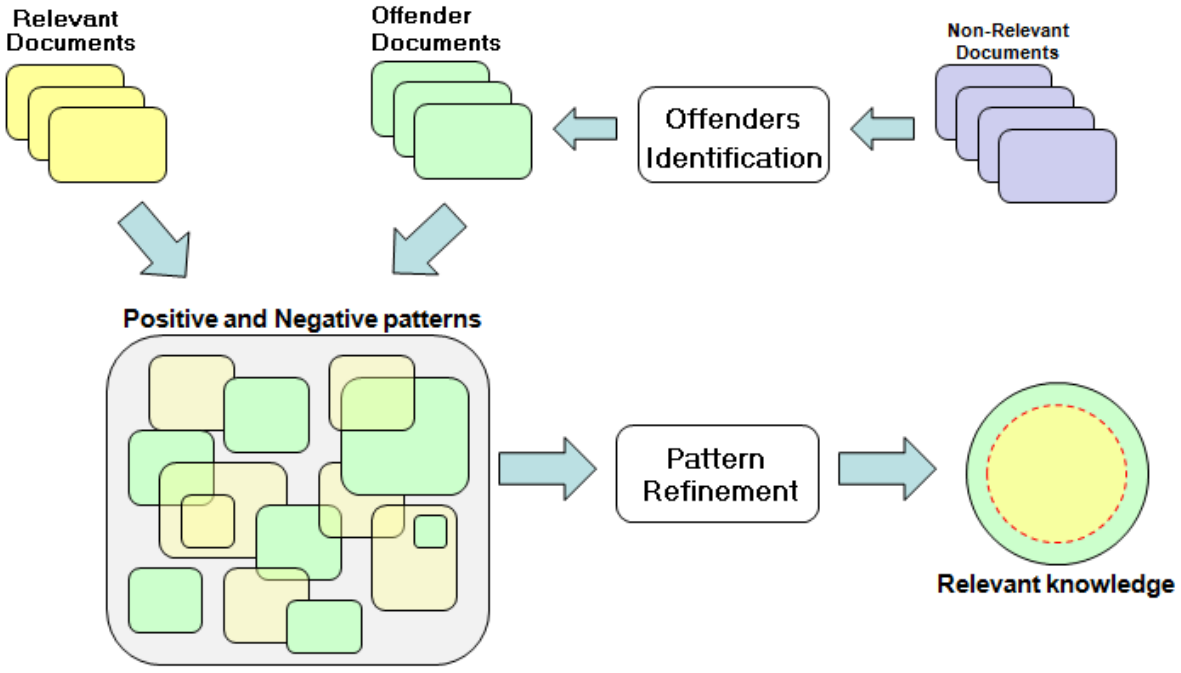



FIGURE 5.2: Pattern cleaning method

### 5.3.1 Offenders Identification

In general, the number of non-relevant documents in a feedback collection may be quite large as compared with the number of relevant ones. This is because

they are easily gathered. However, the discovery of patterns in all the non-relevant documents are not interesting and may increase noise due to the diverse characteristics of non-relevant documents.

Instead of all non-relevant documents, we propose to identify a subset of interesting non-relevant documents used for the updating process. We defined that a non-relevant document is *interesting* if it shares pieces of information about relevant documents. Such a document is useful for reducing ambiguous information and also contains useful non-relevant information (features) for reducing a mistake decision of non-relevant documents. In this context, we call the interesting documents as *offenders*. Figure 5.2 illustrates the area of offenders. According to

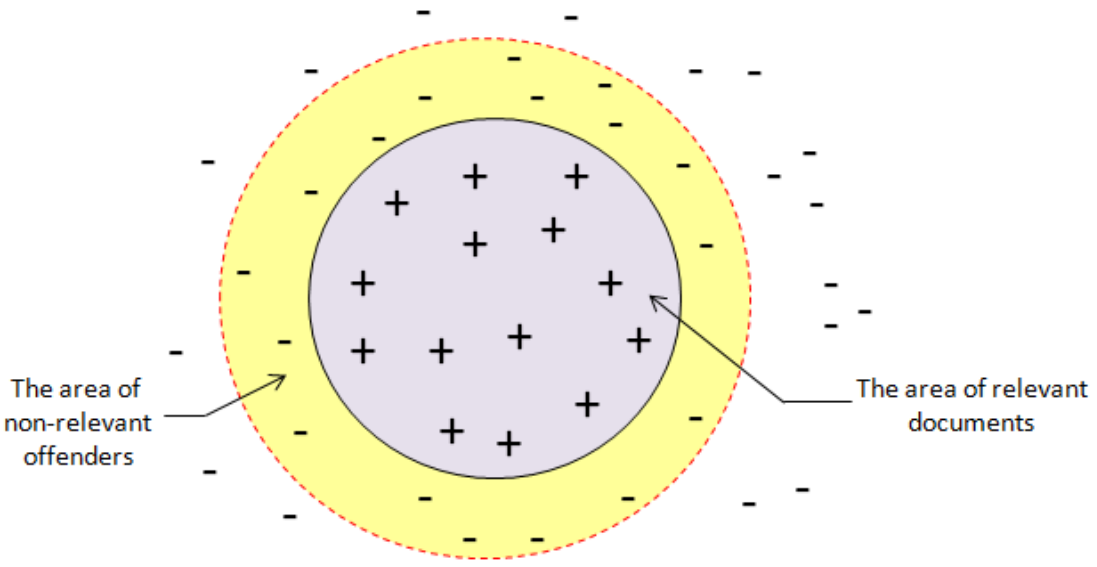



FIGURE 5.3: The area of offenders

this figure, the offender set in the yellow area contains non-relevant documents that close to relevant ones (magenta area) since they share some information each other. The rest of non-relevant documents can be removed.

To identify the offenders, we propose a document evaluation function for scoring non-relevant candidates in $D^-$. Basically, the scoring function assigns a weight to each non-relevant document of a feedback collection based on the occurrence of relevant information in the document. However, matching all patterns in a large document can be considered as an expensive operation. Thus, an efficient method needs to be applied. Given $T^+$ be a set of all frequent terms (size-1 sequential patterns) in relevant documents $D^+$ and a non-relevant document $nd \in D^-$, the weight of $nd$ can be assigned according to the following function:

$$S(nd) = \frac{\sum_{t_j \in nd \cap T^+} tf_j}{\sum_{t_k \in nd} tf_k} \tag{5.1}$$

where $tf_j$ and $tf_k$ denote the term frequencies of $t_j$ and $t_k$ in $nd$. The denominator is used to normalize a large document. The high weight assigned to $nd$ means that $nd$ tends to be an offender. Once the weights of document are identified, we sort the non-relevant documents in descending order associated with their weight, i.e., $S(nd_1) \geq S(nd_2) \geq \cdots \geq S(nd_m)$, where $m = |D^-|$. Then, we choose the top$-k$ ranked documents as the offenders instead of the original set of non-relevant documents. The clear merit of the use of $k$ parameter rather than a threshold value is that the parameter $k$ is less sensitive to the statistics on the dataset and so is easier for human users to specify.

To be clear, given $T^+ = \{t_1, t_2, t_3, t_4\}$ be a list of size-1 sequential patterns extracted from relevant documents, we assume that a set of non-relevant documents $D^- = \{nd_1, nd_2, \ldots, nd_5\}$. Table 5.1 illustrates the sorted non-relevant documents with their weight in Eq (5.1).

| Doc. | List of term frequencies | weight |
|---|---|---|
| $nd_1$ | $\{(t_1, 2), (t_2, 3)\}$ | 1.0 |
| $nd_2$ | $\{(t_1, 3), (t_3, 2), (t_4, 7), (t_6, 1)\}$ | 0.91 |
| $nd_3$ | $\{(t_3, 4), (t_4, 2), (t_5, 7), (t_6, 2)\}$ | 0.40 |
| $nd_4$ | $\{(t_4, 8), (t_6, 3), (t_7, 4), (t_7, 4), (t_8, 2), (t_9, 5)\}$ | 0.36 |
| $nd_5$ | $\{(t_6, 1), (t_8, 4), (t_9, 2)\}$ | 0.0 |

TABLE 5.1: The sorted non-relevant documents $D^-$ with their weight

Finally, PTMining algorithm is applied to discover a set of closed sequential patterns for the top-k offenders.

### 5.3.2 The Refinement Strategy

In this subsection, we describe the second step of pattern cleaning, aiming to refine relevant information by using non-relevant information to describe the relevant knowledge. To avoid the ambiguity, we coin the new terms *positive patterns* and *negative patterns* that refer to closed sequential patterns discovered in relevant documents and offenders respectively.

Figure 5.3 illustrates the relationship between positive and negative patterns discovered in a feedback collection. According to Figure 5.3, positive patterns can be considered as pieces of relevant information in the feedback collection while negative patterns represent non-relevant information. However, some of positive and negative patterns may share some information (i.e., terms or patterns) each other. For example, positive pattern $D$ share some terms with negative pattern $H$ or positive pattern $F$ is subsequence of negative pattern $G$. Such common information can be considered as ambiguous (noisy) information, which may hinder the use of patterns for decision making. These patterns can related to the

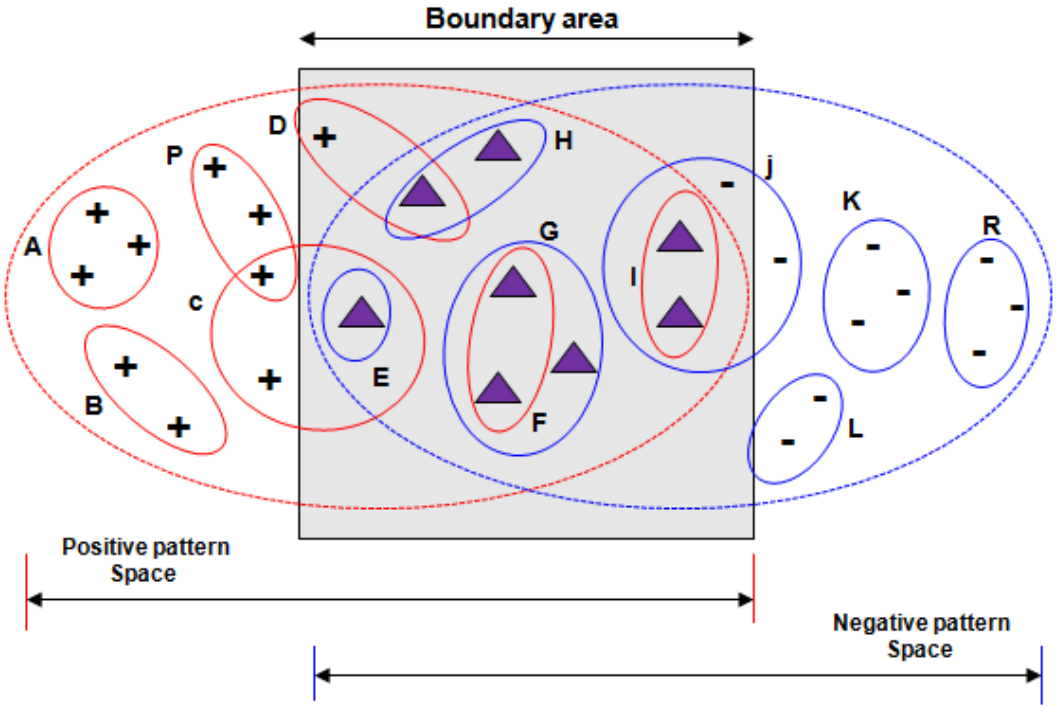



FIGURE 5.4: The relationship between positive patterns (red ovals) and negative patterns (blue ovals) in a training set

boundary area. Depending on the the training dataset, the boundary may be large, which means that it include a lot of ambiguous information. The objective of pattern refinement is to reduce the noisy information by identifying positive and negative patterns necessary to describe the relevant knowledge, especially the boundary between relevant and non-relevant information.

Our proposed method basically classifiers both positive and negative patterns based on their relationship into categories. Then, it deals with the categories of patterns to represent the relevant knowledge. Based on the relationship between positive and negative patterns, we can classify the positive patterns into the three main categories in term of appearance of noisy information: *relevant*, *weak*, and

*conflict*.

We give the formal definitions of the groups of patterns as follows:

*Definition* 7 (Relevant and Weak patterns). A positive pattern $p \in D^+$ is called *weak* if $\exists q \in D^- \Rightarrow p \cap q \neq \emptyset$; otherwise, $p$ is called *relevant*.

Let us consider positive patterns shown in Figure 5.3. Positive patterns $C$, $D$, $F$ and $I$ are weak patterns since we can find negative patterns $E$, $H$, $G$ and $J$ that share some common information, i.e., $C \cap E \neq \emptyset$, $D \cap H \neq \emptyset$, $F \cap G \neq \emptyset$, and $I \cap J \neq \emptyset$.

*Definition* 8 (Conflict patterns). Let $WP$ be the set of weak positive patterns. A positive pattern $p \in WP$ is called *conflict* if $\exists q \in D^-$, such that $p \subseteq q$.

For instance, in Figure 5.3, positive patterns $F$ and $I$ are two conflict patterns since they are the subsets of negative patterns $G$ and $J$, respectively. Conflict patterns are often very short (e.g., size-1 or size-2 patterns) and may be very general to the relevant topic since they tend to frequently occur in both positive and negative data.

Here, we examine some interesting theorems for the noisy patterns:

***Theorem*** 1. Let $PP$ be the set of frequent positive patterns discovered in $D^+$, and $NP$ be the set of frequent negative patterns discovered in $D^-$. A positive pattern $p$ is conflict pattern if and only if $p \in PP \cap NP$.

*Proof.* "⇒", based on Definition 2, there is a negative pattern $q$ such that $p \subseteq q$ if $p \in PP$ is a conflict pattern. Since $q \in NP$ is frequent, and $p$ is a sub-pattern of $q$; $p$ is also a frequent negative patterns. So $p \in PP \cap NP$.

"⇐", if $p \in PP \cap NP$, then $p \in NP$ and $p \subseteq p$. So $p$ is a conflict pattern based on Definition 2. □

***Theorem*** 2. (Anti-Monotonic Property) Let pattern $p \in D^+$ is a conflict pattern. $\forall q \subseteq p$, $q$ is conflict pattern.

*Proof.* Based on Theorem 1, $p$ is a conflict pattern $\Leftrightarrow p \in PP \cap NP$, that is, $q \subseteq p \Rightarrow q \in PP \cap NP$. So, $q$ is a conflict pattern based on Theorem 1 again. □

Considering the overlap information between positive and negative patterns, we also deal with negative patterns from a training set by classifying them into two categorises: *weak negative* and *non-relevant*.
We give the formal definitions as follows.

*Definition* 9 (Weak negative and Non-relevant Patterns). Let $WP$ be the set of weak positive patterns and $CP$ be the set of conflict positive patterns. A negative pattern $q \in D^-$ is called *weak negative* if $\exists p \in WP - CP$, such that $q \cap p \neq \emptyset$; otherwise, $q$ is called *non-relevant*.

***Theorem*** 3. Let $WP$ and $CP$ be sets of weak patterns and conflict patterns identified in positive patterns. A negative pattern $q$ is non-relevant pattern if and only if $q \cap T' = \emptyset$, where $T' = \{t | t \in p, p \in WP - CP\}$.

*Proof.* "⇒", Assume $q$ is non-relevant pattern, but $q \cap T' \neq \emptyset$. Let $t \in q \cap T'$. Then there is a $p \in WP - CP$ such that $t \in p$. $\therefore t \in q \cap p$ and then $q \cap p \neq \emptyset$, i.e., $q$ is a week pattern based on Definition 3. This is contradictory to the assumption.

"⇐", Assume $q \cap T' = \emptyset$, but $q$ is not non-relevant, i.e., $q$ is a week pattern. Based on Definition 3, $\exists p \in WP - CP$, such that $q \cap p \neq \emptyset$. $\because p \subseteq T'$. $\therefore q \cap T' \supseteq q \cap p \neq \emptyset$, that is $q \cap T' \neq \emptyset$. This is contradictory to the assumption. □

Once these patterns are classified, we use the groups of patterns to describe the relevant knowledge of user feedback as shown in Figure 5.4.

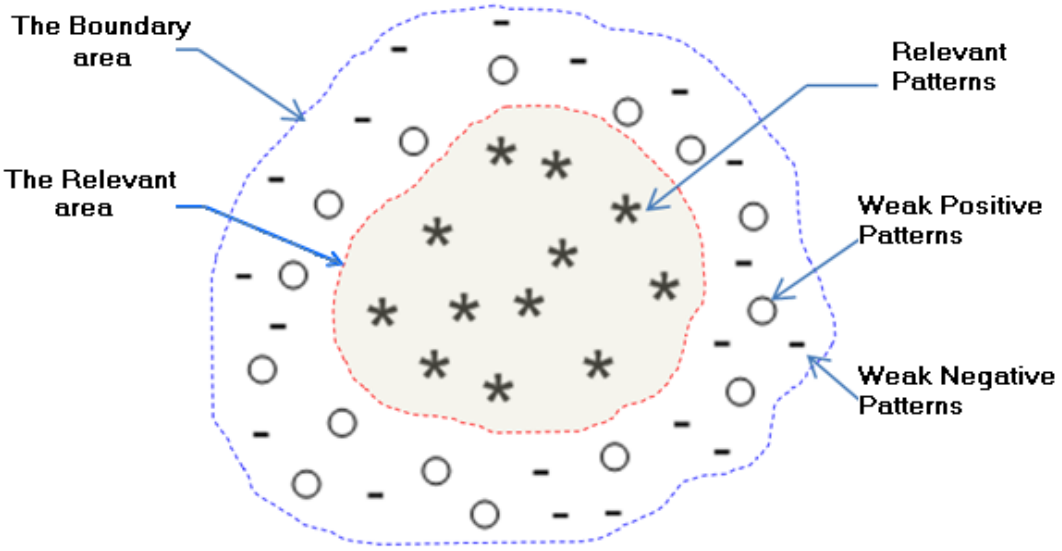



FIGURE 5.5: The relevant knowledge represented by groups of patterns

According to this figure, the relevant knowledge is described by sets of relevant patterns and both weak patterns, including weak positive and weak negative ones. The rest of patterns (i.e., conflict patterns and non-relevant patterns) are removed.

### 5.3.3 Pattern Cleaning Algorithm

Algorithm 3 describes the process of pattern cleaning. The input of the algorithm is a set of positive patterns $PT_{D^+}$ discovered from relevant documents $D^+$, a set

of non-relevant documents $D^-$, and the number of offenders $k$. The output is the updated sets of positive and negative patterns used for describing the relevant knowledge of user feedback.

According to Algorithm 3, this algorithm starts to identify the set of offenders $D^-_{off}$ from non-relevant documents $D^-$ (Steps 1 to 5). Basically, it scores the non-relevant documents using Equation (5.1) and then only the top$-k$ scored documents is chosen as the offenders. Step 6 calls $PTMining$ algorithm that results in the set of closed sequential patterns $PT_{D^-}$ for the identified $k$ offenders. After that, the process of refinement is applied to the discovered sets of positive and negative patterns (Steps 7 to Steps 13). First, it performs to remove conflict patterns with a top-down search strategy (Step 7 to 10), which uses the advantage of anti-monotonic relation property to improve the efficiency (see Property 2). It starts from negative patterns $q$ in the top list of $PT_{D^-}[m]$ to examine positive patterns in $PT_{D^+}$. For each positive pattern $p \in PT_{D^+}$ that is identified as the subset of $q$, we update the pattern profile $PT_{D^+}$ by removing $p$ and its all sub-patterns (Step 10). Steps 11 to 13 describes the process of updating negative patterns in the profiles $PT_{D^-}$ by pruning non-relevant patterns that never overlap with any positive patterns in the updated tree $PT^+$. The output of the algorithm is to yield the updated trees $PT^+$ and $PT^-$ respectively.

#### 5.3.3.1 Example: Pattern Cleaning

Let us show an example of the pattern cleaning algorithm. Assume that pattern taxonomies of positive and negative patterns have already mined from relevant documents $D^+$ and offenders $D^-_{off}$ respectively as shown in Figure 5.6.

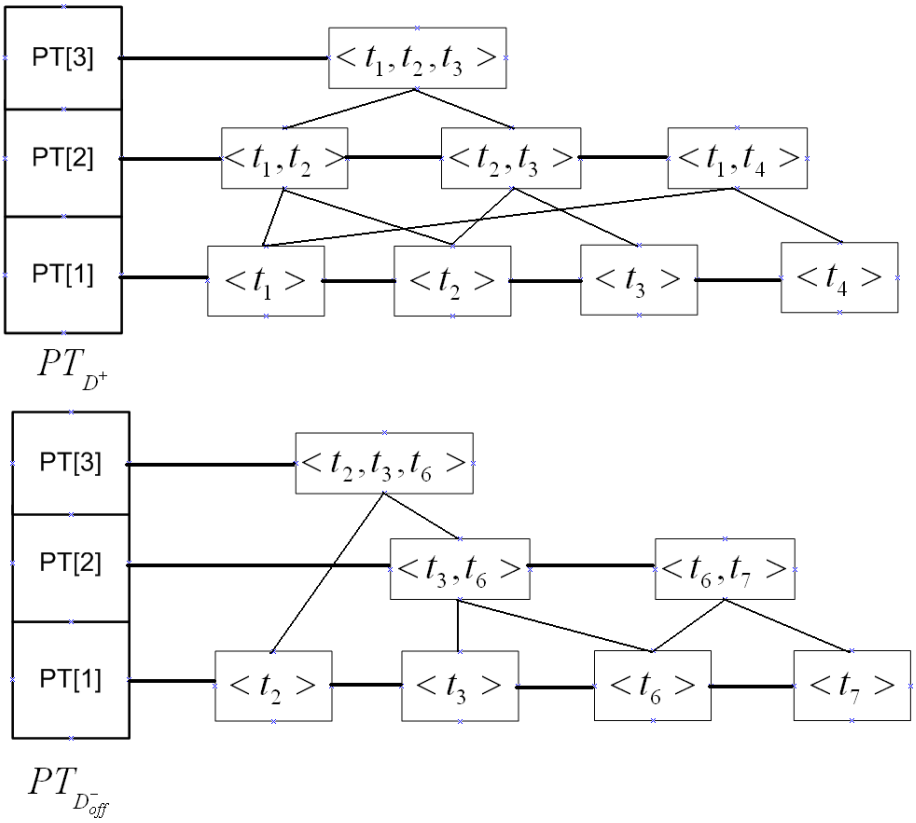



FIGURE 5.6: Pattern taxonomies of positive patterns (top) and negative patterns (bottom)

According to Algorithm 3, the first step of pattern cleaning is to eliminate conflict patterns in a set of positive patterns by matching all positive patterns with negative ones. Once it is found that a positive pattern is subset of a negative pattern, it performs to remove the positive pattern and its all sub-patterns from pattern taxonomy $PT_{D^+}$. Figure 5.7 illustrates the result of pruning conflict patterns $< t_2, t_3 >$, $< t_1 >$, and $< t_2 >$ in the set of positive patterns with respect to negative pattern $< t_2, t_3, t_6 >$.

After conflict patterns are removed, it is easy to identify relevant patterns and

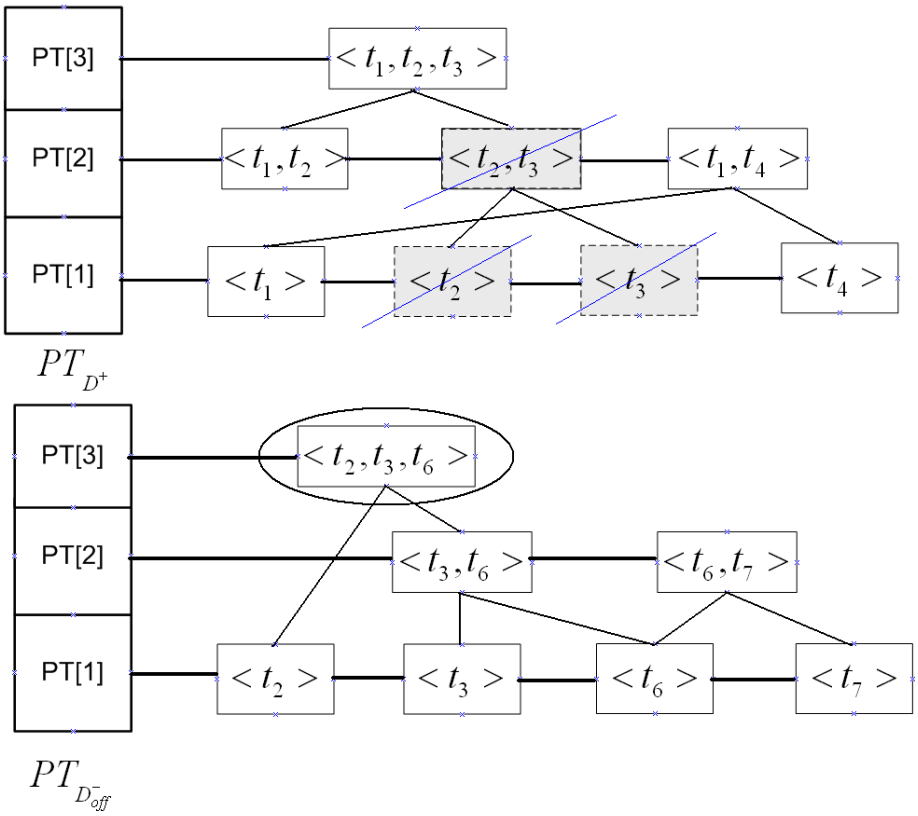



FIGURE 5.7: The result of removing conflict patterns in a set of positive patterns with respect to negative pattern $< t_2, t_3, t_6 >$

weak positive patterns based on the above definitions. Figure 5.8 illustrates the identification of groups of relevant patterns and weak positive patterns. According to this figure, positive patterns $< t_1, t_2, t_3 >$ and $< t_1, t_2 >$ are identified as weak positive patterns since they share common term $t_2$ with negative patterns $< t_2 > < t_2, t_3, t_6 >$. Positive patterns $< t_1 >$, $< t_1, t_4 >$ and $< t_4 >$ are relevant patterns since they never overlap with any negative one. Finally, it uses information (terms) from all weak positive patterns identified to remove non-relevant

patterns and to identify weak negative patterns. Figure 5.9 illustrates the result of removing non-relevant patterns. The rest of negative patterns are weak negative patterns.

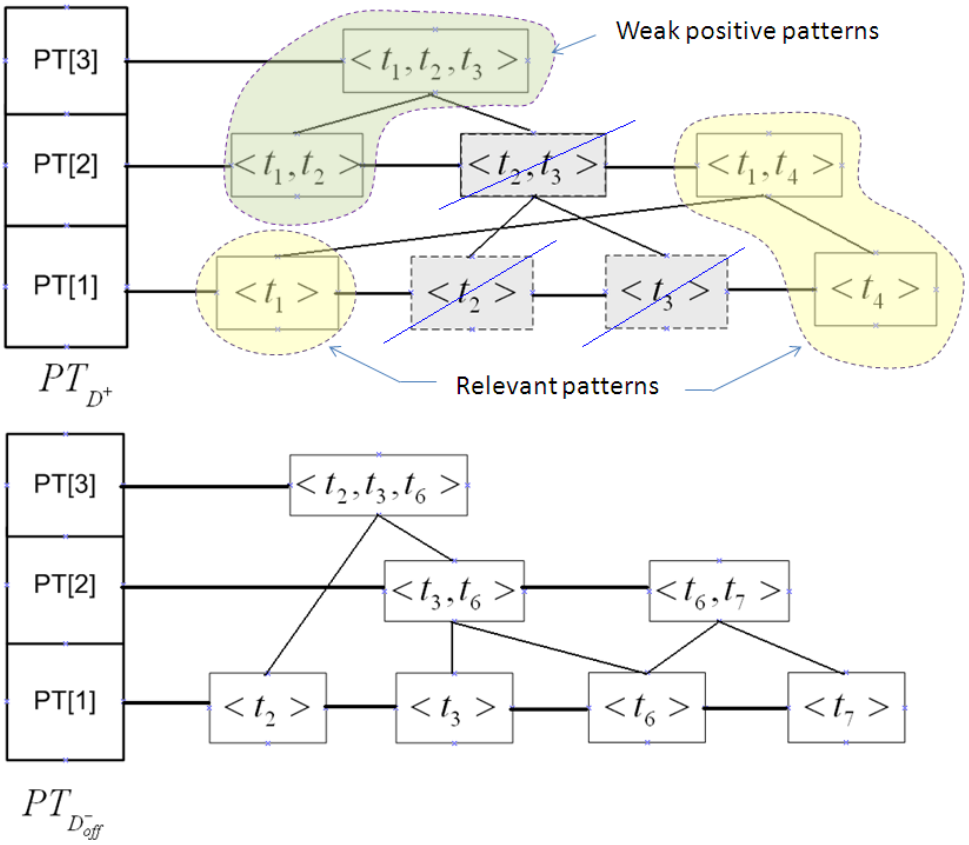



FIGURE 5.8: The identified groups of relevant patterns and weak positive patterns

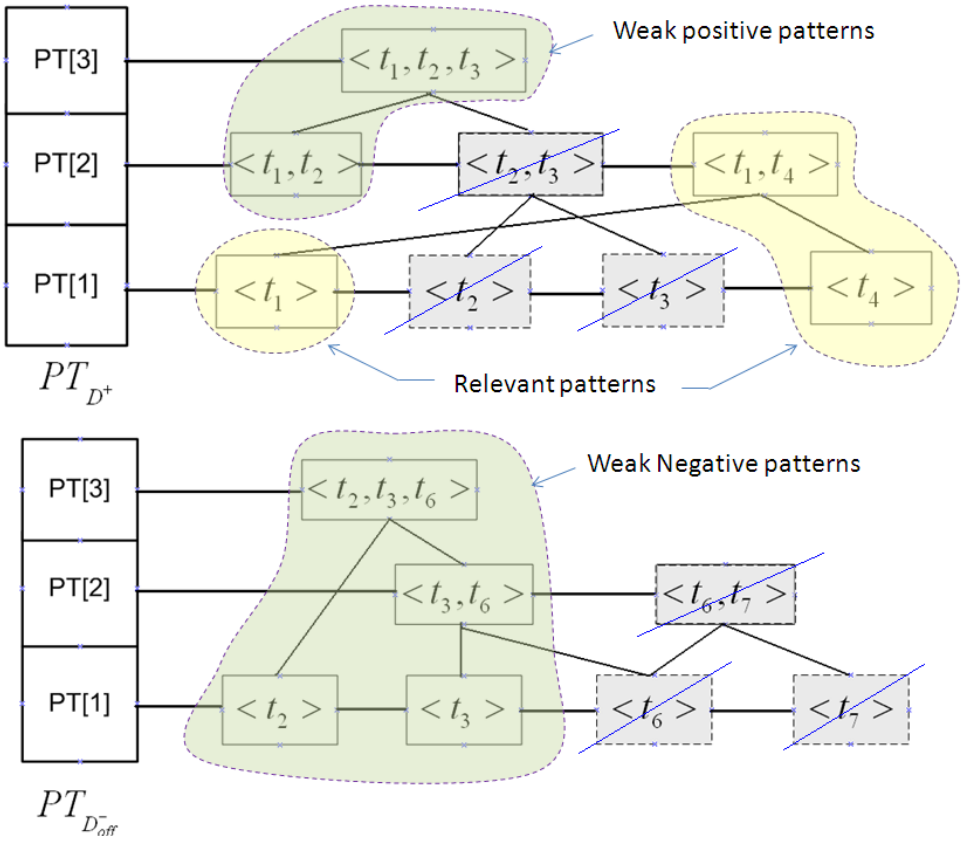



FIGURE 5.9: The result of removing non-relevant patterns in a set of negative patterns

## 5.4 Chapter Summary

This chapter presents the proposed approach for noise reduction in discovered patterns in a feedback collection. This approach focuses on mining both relevant and non-relevant information to precisely describe the relevant knowledge of user feedback. Furthermore, we developed a novel pattern refinement method, called *pattern cleaning* to capture this goal. Based on interesting theorems, the proposed method of pattern cleaning is efficient and scalable due to the desired property of anti-monotonicity.

**Algorithm 3**: Pattern Cleaning Algorithm

**Input** : The pattern taxonomy $PT^+$; Negative documents $D^-$; Number of offenders $K$

**Output**: The updated taxonomies of positive patterns $PT^+$ and negative patterns $PT^-$;

**begin**

1 $T = \{t | t \in p, p \in PT^+\}$;
2 **for** *each negative document* $nd \in D^-$ **do**
3 compute the score $S(nd)$ according to Eq.(5.1);
4 Let $D^- = \{nd_1, nd_2, \ldots, nd_r\}$ in descending order;
5 $D^-_{off} = \{nd_i | nd_i \in D^-, S(nd_i) > 0, i \leq K\}$;
6 $PT^- = PTMining(D^-_{off}), min\_sup)$;
7 **for** $i = |PT^+|$ **to** 1 **do**
8 **for** *each positive pattern* $p \in PT^+[i]$ **do**
9 **for** $j = |PT^-|$ **to** 1 **do**
10 **for** *each negative pattern* $q \in PT^-[j]$ **do**
**if** $p \subseteq q$ **then** Remove $p$ and its children from $PT^+$
11 $T' = \{t | t \in p, p \in PT^+\}$;
12 $PT^- = PT^- - \{p | p \in PT^-, p \cap T' = \emptyset\}$;
13 **return** $PT^+$ *and* $PT^-$*;*

**end**

# Chapter 6

# Relevance Feature Models

In the previous chapter, we present a novel anti-monotone pruning algorithm for mining relevant patterns by removing non-interesting patterns in the collections of positive and negative patterns. In other words, this algorithm results in subsets of positive and negative patterns to describe the relevant knowledge of user's interests. In this chapter, we target on the issue regarding how to utilise the identified subsets of patterns for the effectiveness of relevant feature discovery. To realise this, we develop two relevance feature models for the use of (relevant) patterns which include a set of positive patterns and negative ones to enhance the performance of information filtering, a system that monitors a steam of incoming documents to filter out non-relevant documents with respect to user profiles [11]. In the first model, we focus on the use of extracted relevant patterns as a new feature space for describing the user's relevant documents. Alternatively, the second model has been developed based on the pattern deploying approach [99] which attempts to solve the limitations of using specific long patterns in text

documents.

In addition, we aim to use the proposed relevance models for the evaluation of of the proposed pattern cleaning method comparing with existing approaches to relevant pattern mining. We will describe this in Chapter 7.

## 6.1 Weighted Support Model

Generally, the straight forward way to use discovered patterns in data is to treat patterns as a feature space for building a global model. For example, PTM models [98] that use weighted (closed) sequential patterns as profiles to score documents according to the user's interest. An approach for mining discriminative patterns as a feature space to improve the classification performance was proposed in [17, 22].

In this model, weights are computed and assigned to each discovered pattern to reflect its significance in a training dataset. In document filtering, the similarity between a user's profile and each incoming document is calculated to retrieve documents relevant to the profile. In this case, patterns can be used to evaluate this similarity.

To assess a pattern's significance, statistical measures used in data mining such as "support" or "confidence"[32] can be applied. However, there are two major issues regarding the use of data mining measures for answering what users want [60, 112]. The first issue is *low frequency*. Figure 6.1 illustrates the support distribution of closed sequential patterns in a training document set. According to this figure, the major part of patterns are rare in covering the training documents

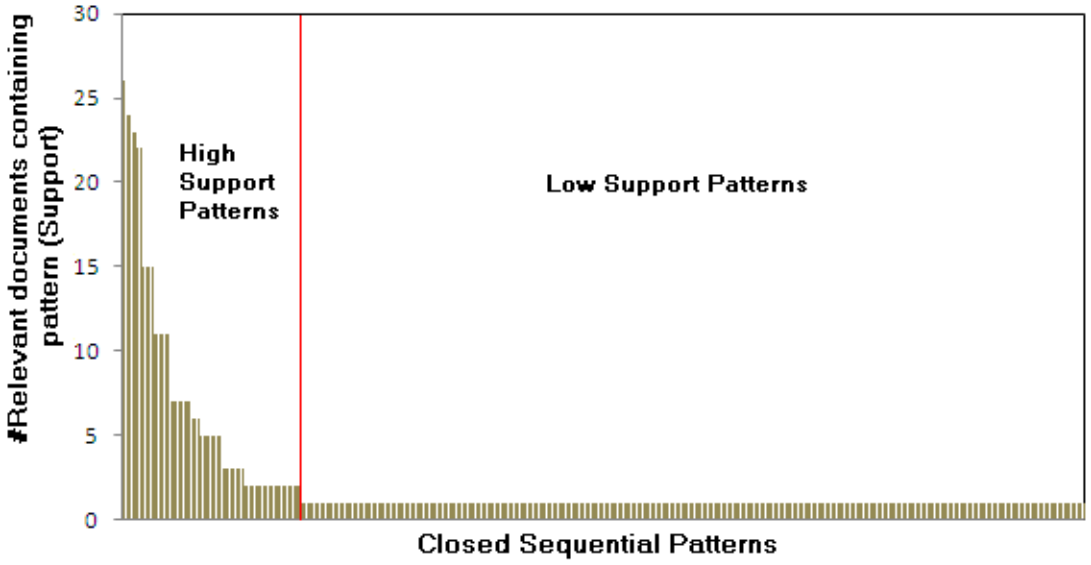



FIGURE 6.1: The support distribution of closed sequential patterns in a training set of documents

with very low support, but a few patterns with high support. The low support patterns can lead to model over-fitting and may weaken the discriminative power of the model [17].

The second issue is *misinterpretation*, which means that a highly frequent pattern (normally a short pattern with large support) is usually a general pattern, but a specific long pattern with low support. The hard problem is how to accurately evaluate the weights of patterns to help distinguish the relevance of documents. As the weights should reflect their relevance, a high weight should be promoted to a long pattern which usually carries more specific information than a shorter one. For instance, comparing pattern "us president" and pattern "us" the former is obviously more meaningful than the latter to identify the relevant documents contained by these patterns. However, the long pattern is sometimes unreliable

since it covers too few documents. We performed a number of experiments, and found that the combination of the support of a pattern and its length performs the best. Given a set of relevant documents $D^+$, the weight of a given pattern $\alpha$ can be calculated as follows:

$$w(\alpha, D^+) = \frac{|\alpha|}{|\alpha| + 1} \times Sup(\alpha, D^+) \tag{6.1}$$

where $Sup(\alpha, D^+)$ denotes the (relative)support of $\alpha$ and $|\alpha|$ is the number of terms contained in $\alpha$. According to the above weighting function, the two pieces of information about each pattern to be combined. The first one is the support of pattern that reflects the reliability of pattern in the training data while the second one aims to emphasize the specificity of pattern. The specificity of pattern is calculated as the normalisation of the length of pattern (i.e., $\frac{|\alpha|}{|\alpha|+1}$).
Generally, there are two types of patterns, including a set of positive patterns $RP^+$ and a set of negative patterns $RP^-$, used for scoring each incoming document in the testing phase. We perform the following steps in the testing phase.

- *Step 1*: Given each incoming document $d$, the set of closed sequential patterns $SP$ that occurs at least two paragraphs of the document are extracted.

- *Step 2*: Given the set of patterns $SP$, the following scoring function is applied to estimate the relevance score with respect to the document.

$$r(d) = \frac{\sum_{\alpha_i \in SP} W(\alpha_i, RP^+) - \sum_{\alpha_j \in SP} W(\alpha_j, RP^-)}{|D^+|} \tag{6.2}$$

where $W(\alpha_i, RP^+)$ indicates the weight of pattern $\alpha_i$ obtained from positive patterns $RP^+$.

The rational behind the above general scoring function is that when there are many positive patterns that can cover the document, it should be assigned a high score so that it will appear in the top positions. Conversely, whenever no or few positive patterns can cover the document, the negative patterns that cover the document results in a low score assigned to the document so that it will be suppressed to the bottom positions.

## 6.2 Extended Pattern Deploying Model

Although patterns have much potential to represent relevant concepts in the training set of documents, previous experiments do not support their use to improve significant performance of text mining in comparing with traditional term-based approaches [24, 87, 98, 112]. The main reason is that many discovered patterns are too specific to be matched in a document, especially specific long patterns.

Recently, an effective method for using closed sequential pattern in text, called *pattern deploying* method [99], has been successful to overcome the disadvantage of pattern mining. The main idea of the deploying method is to use closed sequential patterns discovered in a training set of documents to accurately evaluate the supports(weights) of low-level terms based on their distributions in the patterns. This method has been successfully applied in pattern-based approaches to information filtering [60, 112, 113].

However, we hypothesize that the quality of discovered knowledge (features) obtained by the deploying method has been limited. The first reason is that the data mining method mainly focuses on mining relevant information by ignoring non-relevant one. We believe that both relevant and non-relevant information are necessary and useful for precisely describing the user's interest topic, and finally make the better performance. The second reason is that the deploying method uses simple closed sequential patterns which usually include a lot of noisy information (see theoretical analysis in Chapter 5). Such noise information may affect the correctness of term weights.

Our objective is to extend the deploying method for using both positive and negative patterns in the extraction of high-quality features.

### 6.2.1 Deploying Positive and Negative Patterns

The proposed method of pattern deploying is mapping both positive and negative patterns into a set of low-level features. Basically, the feature set consists of two kinds of features: *positive features* used for describing the user specified relevant documents and *negative features* used for reducing the mistaken decision of non-relevant documents that close to the user interest topic. Figure 6.4 illustrates the deploying of positive and negative patterns.

According to this figure, the deploying of positive and negative patterns results in the extraction of low-level features. However, some of these features may have overlaps between the discovered patterns. Such overlapping features can be considered to be ambiguous due to their use in non-relevant documents.

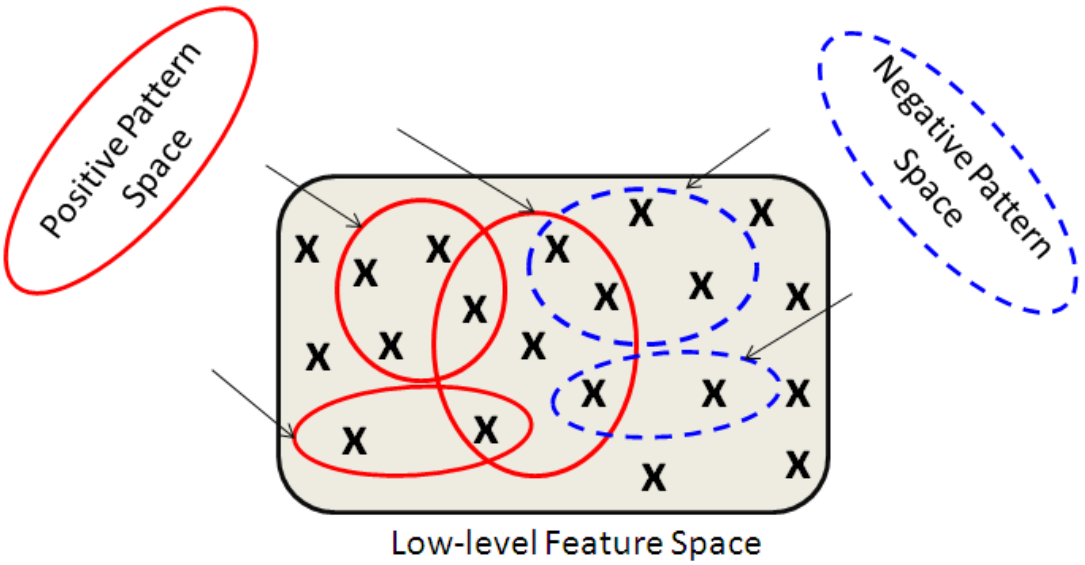



FIGURE 6.2: Mapping positive and negative patterns

The simplest way to avoid the interference by the extracted noisy features is to eliminate all the overlapping features. However, this may cause degrading performance of the system. The first reason is that the number of overlapping features can be large. According to [83], text categorization and filtering still have benefits from large vocabulary. The second reason is that these common features are related in some ways to relevant documents. Thus, they are still important to be used for describing the relevant documents. The appropriate way to reduce the effect of extracted noisy features is to bias their weight based on their distributions in the positive and negative patterns. More specifically, the overlapping features should be lower weighted than non-overlapping ones.

### 6.2.2 The Extended deploying Strategy

Based on analysis in the previous section, we propose the extended deploying strategy for positive and negative patterns. Assume that a set of positive and negative patterns have already been mined from relevant documents $D^+$ and non-relevant documents $D^-$ (or non-relevant offenders) respectively. The following steps describe this deploying.

**S1:** Identify a common set of positive patterns in each relevant document. Let $R_i$ be a set of common patterns in a relevant document $d_i \in D^+$, where $i = 1, 2, \ldots, |D^+|$.

**S2:** Extract a set of positive features $T^+$ in relevant documents $D^+$ using the deploying method using Eq.(6.4).

**S3:** Extract a set of negative features $T^-$ in non-relevant documents $D^-$ in the same way of positive features.

**S4:** Normalize the term supports in $T^+$ and $T^-$ obtained in Steps 2 and 3 respectively according to the following equation.

$$w_{norm}(t, T^i) = \frac{w(t, T^i)}{\sqrt{\sum_{t_j \in T^i} w(t, T^i)}} \tag{6.3}$$

where $w(t, T^i)$ denotes the deployed weight of term $t$ in feature set $T^i$, $i \in \{+, -\}$.

**S5:** Let $T = T^+ \cup T^-$, The following function is used to update the term supports (weights).

$$w_\Delta(t) = \begin{cases} w(t, T^+) & ; t \in T^+ - T^- \\ w(t, T^+) - w(t, T^-) & ; t \in T^+ \cap T^- \\ -w(t, T^-) & ; t \in T^- - T^+ \end{cases} \tag{6.4}$$

for all terms $t \in T^+ \cup T^-$.

The basic idea of weight update is to reduce the weights of common features in $T^+$ and $T^-$, which may interfere the correct decision of relevant documents. We also update the weights of negative features which are typically positive by altering their weight from positive to negative value.

In order to use the extracted low-level features, we build a document evaluation function for scoring a test document based on its relevance. Given a test document $d$, the relevance score of the document is calculated by the following function.

$$r(d) = \sum_{t_i \in T} w_\Delta(t_i) \times \tau(t_i, d) \tag{6.5}$$

where $w_\Delta(t_i)$ is the weight of feature $t_i$ and $\tau(t_i, d) = 1$ if $t \in d$; otherwise $\tau(t, d) = 0$. A high value assigned to the document can imply that the document is highly relevant.

## 6.3 Chapter Summary

This chapter presents two effective models for using the relevant knowledge including a set of positive and negative patterns in a feedback set of documents. In the first model, the mined patterns in the phase of pattern discovery are treated as a feature space to precisely describe the user specified relevant documents. A novel pattern evaluation method has also been proposed to determine the pattern weights for use in the decision of relevance.

The second model with the attempt of addressing the difficulties in using specific long patterns in text documents has also been proposed. This model deploys the positive and negative patterns into a weighted vector of low-level terms which are easily matched in a document. The supports of terms are evaluated based on their appearance in these patterns.

The attraction of the proposed models of relevance is that these models can be easily applied to use different kinds of knowledge patterns. This allows to evaluate the quality of discovered knowledge obtained by using different pattern mining methods. The evaluation results will be presented in Chapter 7.

# Chapter 7

# Experiments and Results

This chapter describes experimental evaluations of the proposed framework. Two main hypotheses have been proposed in this research. These two hypotheses are:

- A post-processing method for frequent patterns in text is necessary to improve the quality of extracted features for describing user information needs or preferences.

- The pattern cleaning method is useful for reducing the noise in discovered patterns from positive feedback documents.

To evaluate the proposed hypotheses, this chapter discusses the testing environment including the dataset, baseline models, and evaluation methods. For the first hypothesis, we report the results and the discussions for the following main points: (1) the proposed approach is significant compared to the baseline models

based on effectiveness and (2) the effectiveness of using post-mining methods to reduce noisy patterns from positive feedback documents is significant.

For the second hypothesis, the pattern cleaning method is significant compared to other post-mining methods in efficiency and effectiveness. We also provide more results and discussions about different offender identification and the effects of selecting different subsets of positive and negative patterns on the effectiveness. Two popular post-mining methods in data mining, including *DPMine* [17] and *Emerging patterns* [22], are employed as baseline models for the purpose of evaluation. The main reason is that these data mining methods focus on seeking the patterns that are relevant to the class of interest (i.e., positive feedback documents).

In this thesis, a practical task of information filtering (IF) has been conducted to evaluate the two proposed hypotheses. The TREC-11 Reuter's corpus is chosen as our benchmark collection for the experiments. Effectiveness was determined by both standard information retrieval/filtering measurements and statistical significant different measurement, i.e., the paired t-test.

## 7.1 Experimental Dataset

The most frequently used collection for experiments in text categorization and filtering area is the Reuters dataset. Over the years, several versions of Reuters corpora, such as Reuters-21578 [40], OHSUMED [42], and 20 Newsgroups collections [52], has been released. Among the common data collections, Reuters Corpus Volume 1 (or *RCV1*) [55] has been the most commonly used dataset for

the experimental evaluation. The RCV1 dataset with roughly 1 GB consists of about 800,000 documents of Reuters new articles during a 1-year period between August 20, 1996, and August 19, 1997. Some key statistics of the RCV1 dataset are shown in Table 7.1.

| **Statistic** | **Value** |
|---|---|
| The total number of documents | 806,791 |
| The total number of paragraphs | 9,822,391 |
| The total number of terms | 96,969,056 |
| The total number of (distinct) terms | 391,523 |
| The average number of unique terms in a document | 75.70 |
| The average document length | 123.90 |

TABLE 7.1: The key statistics of RCV1 data collection [55]

All the documents in the RCV1 dataset are prepared in the XML format with some meta-data information. A typical XML document in RCV1 dataset is shown in Figure 7.1.

According to Figure 7.1, each document is identified by a unique item ID and corresponded with a title in the field marked by the tag $< title >$. The main content of the document is in a distinct $< text >$ field consisting of one or several paragraphs. Each paragraph is enclosed by the XML tag $< p >$. In our experiment, both the "title" and "text" fields are used and each paragraph in the "text" field is viewed as a transaction in a document database. Moreover, we treat the content in the "title" field in the document as an additional paragraph (i.e., transaction). As a consequence, each RCV1 document contains at least two paragraphs, i.e., one for its title and one for its content.

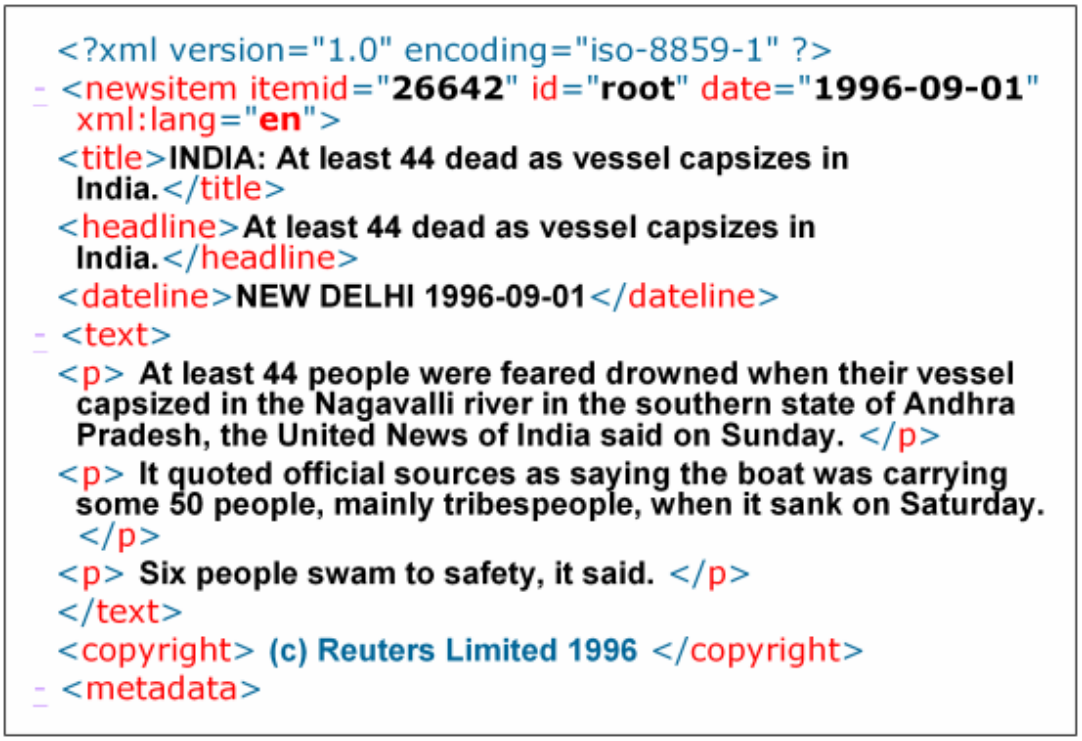



FIGURE 7.1: An XML document in RCV1 dataset

The distribution of paragraphs in the RCV1 training documents are shown in Figure 7.2. As seen in this figure, the documents in RCV1 dataset contain multiple paragraphs. The majority of the RCV1 documents contain at least three paragraphs while the large population include between 3 and 23 paragraphs. The characteristic of multiple paragraphs in RCV1 documents allows frequent pattern mining to be potentially applied at the level of paragraph.

The RCV1 dataset also contains 100 topics, which cover a wide range of international topics, including politics, business, sports, and science. Each topic in the RCV1 dataset contains a reasonable number of documents with relevance judgement both in the training and testing examples.

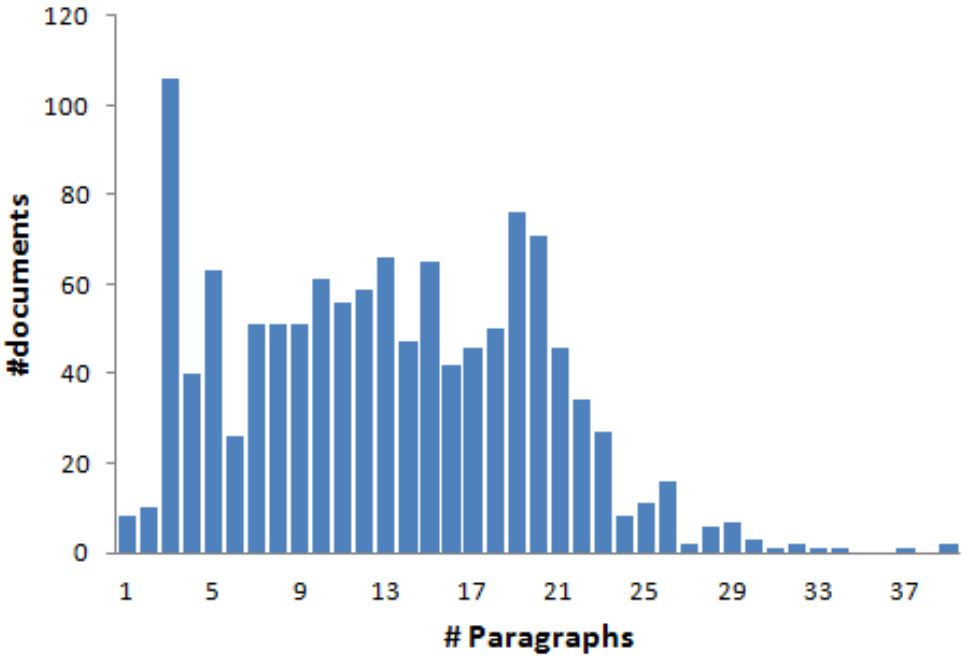



FIGURE 7.2: The distribution of paragraphs in RCV1 documents

In general, the topics in RCV1 are of two types:

- 1) *Assessor topics*: The first set of 50 topics (R101 to R150) are developed by humans and researchers of the National Institute of Standard and Technology (NIST). The relevance judgements have been also made by the assessor of NIST.

- 2) *Intersection topics*: The second set of 50 topics (R151 to R200) have been constructed artificially from the intersections of pairs of Reuters categories. Different from the assessor topics, the relevance judgements have been made by a machine-learning method, not by human beings.

The 50 assessor topics are typically more realistic and reliable than the intersection topics [60]. Each RCV1 topic was also divided into two sets: training set

and test set, where relevance judgements have been proposed for each document in each topic. Table 7.2 illustrates the statistical information about topic R101 to R150.

As seen in Table 7.2, there are 23 documents in the trainings set of topic 101, where 7 of them are positive (relevant) documents $D^+$ used for building a user profile at the training phase. For evaluating phase, there are 577 documents in the testing set.

We also found that the average number of negative (non-relevant) documents in the RCV1 topics are three times as much as that number of positive ones (i.e., 12.78). Furthermore, the majority of testing documents has been dominated by negative ones. Obviously, the characteristic of the real-world topics are the highly imbalanced datasets, which may hinder data analysis [88]. Further details regarding the RCV1 dataset can be found in [55].

In the experiments, we use RCV1 and the 50 assessor topics (from topics 101 to 150) to evaluate the proposed system because the TREC topics are realistic and reliable. Furthermore, according to the experiments in [13], a stable and sufficient evaluation for a retrieval system should be taken into account of at least 50 different topics or queries.

## 7.2 Performance Measures

To evaluate the proposed system, several standard precision/recall measurements in IR are used [8]. The *precision* is the fraction of retrieved documents that are

| Topic ID | Training Set | | | Test Set | | |
|---|---|---|---|---|---|---|
| | $\|D\|$ | $\|D^+\|$ | $\|D^-\|$ | $\|D\|$ | $\|D^+\|$ | $\|D^-\|$ |
| 101 | 23 | 7 | 16 | 577 | 307 | 270 |
| 102 | 199 | 135 | 64 | 308 | 159 | 149 |
| 103 | 64 | 14 | 50 | 528 | 61 | 467 |
| 104 | 194 | 120 | 74 | 279 | 94 | 185 |
| 105 | 37 | 16 | 21 | 258 | 50 | 208 |
| 106 | 44 | 4 | 40 | 321 | 31 | 290 |
| 107 | 61 | 3 | 58 | 571 | 37 | 534 |
| 108 | 53 | 3 | 50 | 386 | 15 | 371 |
| 109 | 40 | 20 | 20 | 240 | 74 | 166 |
| 110 | 91 | 5 | 86 | 491 | 31 | 460 |
| 111 | 52 | 3 | 49 | 451 | 15 | 436 |
| 112 | 57 | 6 | 51 | 481 | 20 | 461 |
| 113 | 68 | 12 | 56 | 552 | 70 | 482 |
| 114 | 25 | 5 | 20 | 361 | 62 | 299 |
| 115 | 46 | 3 | 43 | 357 | 63 | 294 |
| 116 | 46 | 16 | 30 | 298 | 87 | 211 |
| 117 | 13 | 3 | 10 | 297 | 32 | 265 |
| 118 | 32 | 3 | 29 | 293 | 14 | 279 |
| 119 | 26 | 4 | 22 | 271 | 40 | 231 |
| 120 | 54 | 9 | 45 | 415 | 158 | 257 |
| 121 | 81 | 14 | 67 | 597 | 84 | 513 |
| 122 | 70 | 15 | 55 | 393 | 51 | 342 |
| 123 | 51 | 3 | 48 | 342 | 17 | 325 |
| 124 | 33 | 6 | 27 | 250 | 33 | 217 |
| 125 | 36 | 12 | 24 | 544 | 132 | 412 |
| 126 | 29 | 19 | 10 | 270 | 172 | 98 |
| 127 | 32 | 5 | 27 | 238 | 42 | 196 |
| 128 | 51 | 4 | 47 | 276 | 33 | 243 |
| 129 | 72 | 17 | 55 | 507 | 57 | 450 |
| 130 | 24 | 3 | 21 | 307 | 16 | 291 |
| 131 | 31 | 4 | 27 | 252 | 74 | 178 |
| 132 | 103 | 7 | 96 | 446 | 22 | 424 |
| 133 | 47 | 5 | 42 | 380 | 28 | 352 |
| 134 | 31 | 5 | 26 | 351 | 67 | 284 |
| 135 | 29 | 14 | 15 | 501 | 337 | 164 |
| 136 | 46 | 8 | 38 | 452 | 67 | 385 |
| 137 | 50 | 3 | 47 | 325 | 9 | 316 |
| 138 | 98 | 7 | 91 | 328 | 44 | 284 |
| 139 | 21 | 3 | 18 | 253 | 17 | 236 |
| 140 | 759 | 11 | 48 | 432 | 67 | 365 |
| 141 | 56 | 24 | 32 | 379 | 82 | 297 |
| 142 | 28 | 4 | 24 | 198 | 24 | 174 |
| 143 | 52 | 4 | 48 | 417 | 23 | 394 |
| 144 | 50 | 6 | 44 | 380 | 55 | 325 |
| 145 | 95 | 5 | 90 | 488 | 27 | 461 |
| 146 | 32 | 13 | 19 | 280 | 111 | 169 |
| 147 | 62 | 6 | 56 | 380 | 34 | 346 |
| 148 | 33 | 12 | 21 | 380 | 228 | 152 |
| 149 | 726 | 5 | 21 | 449 | 57 | 392 |
| 150 | 51 | 4 | 47 | 371 | 54 | 317 |
| **AVG** | **54.08** | **12.78** | **41.30** | **378.02** | **69.68** | **308.34** |

TABLE 7.2: Statistic Information about the RCV1 assessor topics

| | | **Human judgement** | |
|---|---|---|---|
| | | YES | NO |
| **System judgement** | YES | TP (True Positive) | FP (False Positive) |
| | NO | FN (False Negative) | TN (True Negative) |

TABLE 7.3: Contingency table

relevant to a given topic and the *recall* is the fraction of relevant documents that have been retrieved by the retrieval system.

Since the problem can be viewed as a binary classification problem (positive/negative classes), the precision/recall can be defined within a contingency table as shown in Table 7.3. According to this table, the precision ($P$) and recall ($R$) are defined as the following formulas:

$$P = \frac{TP}{TP + FP}, R = \frac{TP}{TP + FN} \tag{7.1}$$

where $TP$ is denoted as the number of documents the system correctly identifies as positives; $FP$ denotes the number of documents the system falsely identifies as positives; and $FN$ is the number of relevant documents the system fails to identify.

By using the above definitions, the following list is effectiveness measures used for experimental evaluation.

- **Top$-k$ precision**: The precision of first $K$ retrieved documents ($top - k$) is adopted for this experiment since most users expect to see what they are looking for on the first few retrieved documents. The value of $K$ we use in this experiments is 20.

- **Break even Point** ($b/p$): this measure indicates the point where the value of precision equals to the value of recall for a topic. The higher the figure of $b/p$, the more effective the system is.

- **F-beta** ($F_\beta$): this measure basically combines precision and recall to assess the effect involving them. The $F_\beta$ measure can be defined by the following equation:

$$F_\beta = \frac{(\beta^2 + 1) \times P \times R}{\beta^2 \times P + R} \tag{7.2}$$

  where $\beta$ is parameter giving weights of precision and recall and can be viewed as the retrieve degree of importance attributed to precision and recall. A value $\beta = 1$ is adopted in our study, which mean that it attributes equal importance to precision and recall. Therefore, $F\_\beta$ is denoted by:

$$F_1 = \frac{2 * P * R}{P + R}$$

- **Mean Average Precision (MAP)**: this measure is calculated by measuring the precision at each relevant document first, and then averaging the precision over all the topics. It combines precision, relevance ranking and overall recall together to measure the quality of the retrieval engines.

- **Interpolated Average Precision (IAP)**: this metric is used to compare the performance of different systems by averaging the precisions at 11 standard recall levels (recall $= 0.0, 0.1, \ldots, 1.0$). The *11-points* measure indicates the first value of the 11 points where recall equals zero.

A statistical analysis method is also used to analyse the experimental results. Statistical data analysis allows us to use mathematical principles to decide the likelihood that our sample results match our hypothesis about a population. In statistical hypothesis testing, a $p-$value (probability value) is used to decide whether there is enough evidence to reject the null hypothesis and that the research hypothesis is supported by the data.

The $p-$value is a numerical statement of how likely it is that we could have obtained our sample data even if the null hypothesis is true. By convention, if the $p-$value is less than $0.05 (p < 0.05)$, we conclude that the null hypothesis can be rejected. In other words, when $p < 0.05$ we say that the results are statistically significant.

The $t-$test is probably the most commonly-used statistical data analysis procedure for hypothesis testing. There are several kinds of $t-$tests, but the most common is the "two-sample t-test" also known as the "Student's t-test". The two-sample t-test assesses whether the mean values of the two groups are statistically different from each other on some measures.

The paired two-tailed t-test is used in this thesis. If $DIF$ represents the difference between the two observations, the hypothesis are: $H_0 : DIF = 0$ (the difference between the two observation is 0); $H_a : DIF \neq 0$ (the difference are not 0). The test statistic is $t$ with $N - 1$ degrees of freedom ($df$), where $N$ is the sample size of group. If the $p-$value associated with $t$ is low ($< 0.05$), there is evidence to reject the null hypothesis. Thus, there is evidence that the difference in means across the paired observations is significant.

## 7.3 Evaluation Procedure

In order to evaluate the proposed framework, the task of information filtering (IF) is applied. As mentioned in *Chapter2*, IF system is a system that monitors a steam of incoming documents, aiming to filter out non-relevant documents according to profiles of user's interests. The objective of this experiments is to construct effective profile models of users to enhance the effectiveness of IF system.

The evaluation process is illustrated in Figure 7.3. This procedure starts by assuming that each assessor topic in the RCV1 dataset is a feedback collection of documents given by each user. As shown in Table 7.2, an RCV1 topic consists of two sets of documents used for training and testing purposes. Thus, the system uses all of the documents in these two sets for the phases of training and testing. In the following subsections, we explain more details about the subsequent processes in the evaluation procedure.

### 7.3.1 Document Preprocessing and Transformation

Once a training set of feedback documents is provided, each document in the training data is pre-processed. Since RCV1 documents are in XML format, there are many fields enclosed by tags including $< title >$, $< headline >$, $< dateline >$, $< text >$, $< copyright >$, and $< metadata >$. In this experiment, only the fields $< title >$ and $< text >$ are chosen to represent the content of document; otherwise it is discarded.

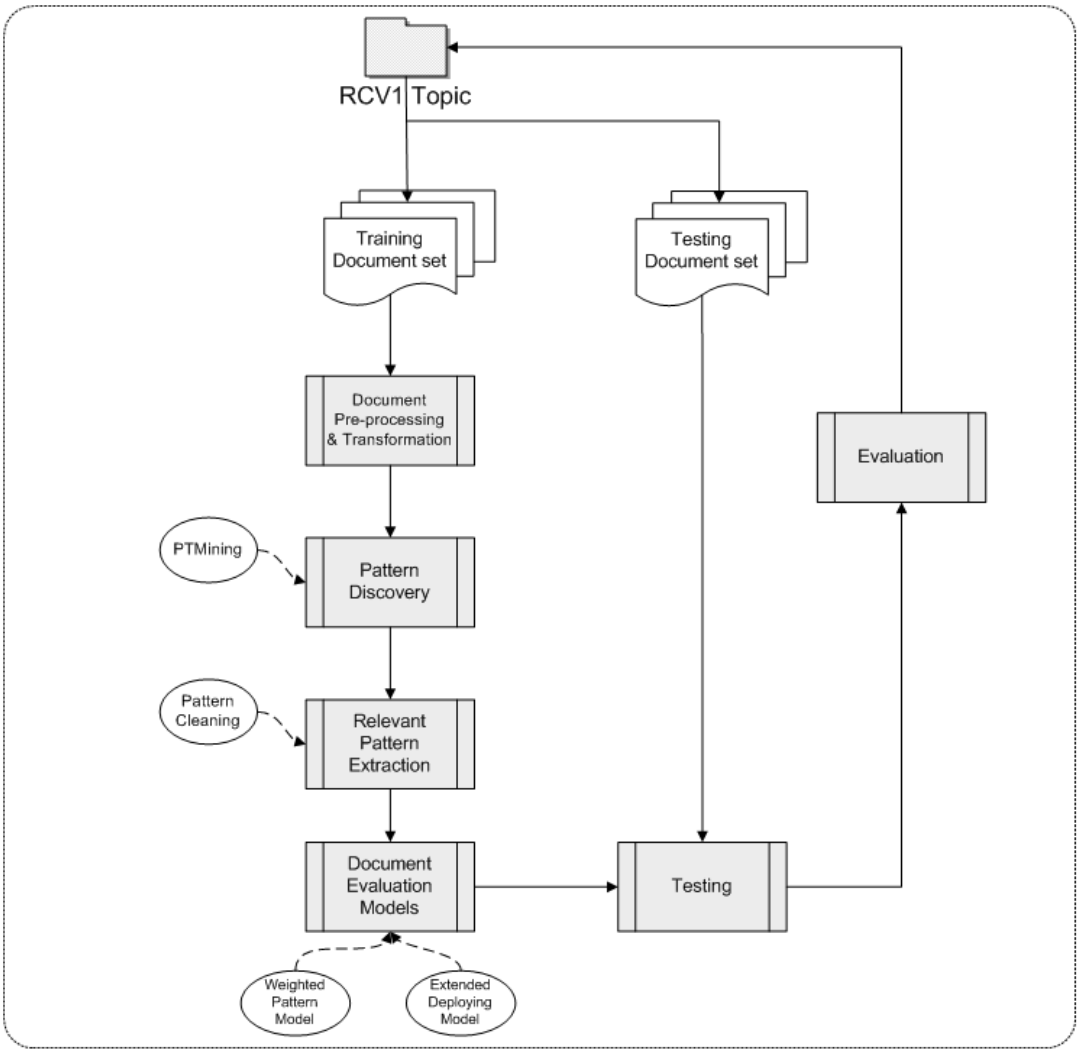



FIGURE 7.3: The evaluation procedure

The next step of document pre-processing is to apply stopword removal and word steaming for reducing noise in a document. Stopwords can be defined as functional words, which helps construct sentences (i.e., articles, prepositions, and conjunctions), or non-informative words (i.e., the words frequently occurring in all English documents). All the stopwords are removed from the document

according to a given stopword list. The stopword list used in this experiment is illustrated in Appendix **??**. The word stemming offers reducing inflected words to their stem or root form in order to reduce the problem of a variety of word forms. In this experiment, the popular word steaming algorithm Porter algorithm [94], is used in this experiment to transform words into its root form.

Once documents were pre-processed, the number of words in the documents can be still quite large. For the purpose of dimension reduction, TF-IDF term weighting scheme (see Chapter 2) is applied to identify the most $k-$informative words for document representation. In this experiment, we set $k = 4,000$ for each assessor topic.

Finally, each processed document is transformed into several paragraphs, where each paragraph contains a sequence of stemmed words as an individual transaction. Figure 7.4 illustrates the process of document preprocessing.

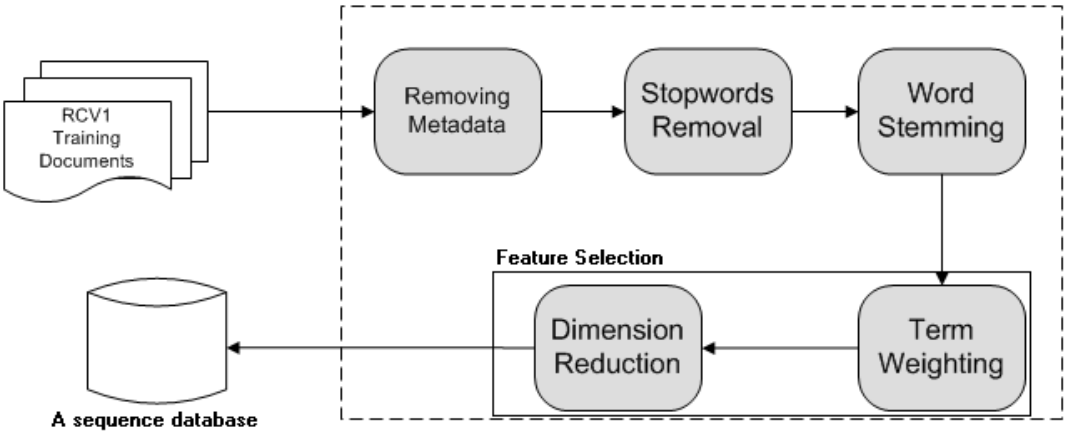



FIGURE 7.4: Document preprocessing

### 7.3.2 Procedure of Pattern Discovery

The result of document preprocessing and transformation is a set of stemmed word sequences in the training documents. The next step is to apply data mining algorithms to extract frequent patterns in the sequence database. As proposed in Chapter 4, PTMining algorithm can be applied to discover frequent (closed)sequential patterns in this database.

Like conventional pattern mining algorithms, PTMining requires users to set a *minimum support threshold (min_sup)* for removing insignificant patterns and noisy patterns accidentally formed. The output of PTMining is a profile of extracted (closed)sequential patterns and their relation in the training dataset.

### 7.3.3 Procedure of relevant pattern extraction

The result of discovered closed sequential patterns obtained by PTMining are passed to the subsequent process to extract relevant patterns and to eliminate noisy ones to improve the effectiveness of pattern mining.

For this purpose, the proposed method of pattern cleaning (PCM) proposed in Chapter 5 is applied in our proposed approach. This method requires both positive and negative patterns discovered from a set of relevant documents and some non-relevant documents (or *offender*) respectively for extracting groups of patterns which consists of relevant patterns and both weak patterns to describe the relevant knowledge in a class (topic) of user's interest.

As mentioned in the previous sections, the discovery of relevant patterns has strong connection to discriminative pattern mining in data mining community.

Thus, existing algorithms for finding discriminative patterns can be adopted to extract the relevant patterns for describing the user's interest topic. For this purpose, we compare the quality of relevant patterns obtained by PCM to the relevant patterns obtained by two-phases algorithms for discriminative pattern mining, including *DPMine* [17] and emerging pattern mining [22] for the effectiveness of information filtering.

### 7.3.4 Document Evaluation

The task of information filtering is to acquire profiles of user interests used for filtering out the documents that are non-relevant to the user's interest.

Once the process of finding relevant patterns were applied, a document evaluation model is built using the extracted relevant patterns. Two relevance ranking models proposed in Chapter 6 can be applied for the use of relevant patterns in the testing phase.

The first model (i.e., weighted pattern model) treats patterns as a feature space for describing the user specified relevant documents. The second model (i.e., extended deploying model) deploys the relevant patterns into a weighted vector of low-level terms with the attempt to address the difficulties of using specific long patterns in text.

### 7.3.5 Testing and Evaluation

For the testing phase, each document in a test set is evaluated by estimating the relevance score of the test document using the document evaluation function

obtained in the training phase. All of the documents in the test set are ranked according to their relevance score.

The system's performance is determined by using several effectiveness measures including top$-k$ precision, $F_{beta=1}$, MAP, $b/p$, and IAP. After evaluation, the system assesses the next topic if required. Finally, the $t-$test statistical significance is also used for evaluating the difference between two systems on the assessor topics.

## 7.4 Baseline Models and Settings

Several approaches to IF are chosen and developed as the baseline approaches compared with our proposed approach. Basically, these approaches can be grouped into two major approaches (1) *data mining-based* and (2) *term-based* approaches. The first approach includes pure data mining-based methods for information filtering. We also group these methods into two main categories. The fist category consists of existing IF methods that use closed sequential patterns, including including PTM [98], PDS [99], and IPE [112]. For the second category, we implemented discriminative pattern-based IF models using DPMine [17] and emerging pattern mining [22].

The second approach includes the popular term-based IR methods, including Rocchio [46], BM25 [75], and ranked-based SVM [112].

### 7.4.1 Data Mining-based Methods

This section describes the details of the data mining-based methods.

| Methods | Relevant Features | Algorithms |
|---|---|---|
| PTM | Weighted Closed Seq. Ptrns. | [98] |
| PDS | Terms with Deployed Weights | [99] |
| IPE | Terms with Deployed Weights | [112] |
| DPMine | Discriminative Patterns | MMRFS [17] |
| EPMine | Emerging Patterns | [22] |
| Rocchio | Low-level Terms | Eq.(7.8) |
| BM25 | Low-level Terms | Eq.(7.9) |
| Ranked-based SVM | Low-level Terms | Eq.(7.10) |

TABLE 7.4: The list of method used for evaluation

- **PTM** [98]: This method uses closed sequential patterns discovered in a set of relevant documents to represent the concept of user interest topic in the training set. To utilise the discovered closed patterns, weights are assigned to each pattern $p$ based on its appearance in a training set of relevant and non-relevant documents as the following equation.

$$w(p) = \frac{|\{d_i | d_i \in D^+, p \in d_i\}|}{|\{d_j | d_j \in D, p \in d_j\}|} \tag{7.3}$$

  where $d_i$ and $d_j$ are training documents in a training set of documents $D$. Given a test document, the weighted patterns are used to estimate the relevance of the document based on the total weight of discovered patterns contained in the document.

- **PDS** [99]: The data mining method for using frequent patterns in text was proposed. This method focuses on addressing the difficulties of using specific long patterns in text by using patterns to weight accurately low-level terms based on their distributions in the patterns. Given a term

$t \in D^+$, the support of term $t$ can be computed as the following function.

$$w(t) = \sum_{i=1}^{|D^+|} \sum_{t \in p \subseteq SP_i} \frac{Sup_a(p, d_i)}{|p|} \tag{7.4}$$

where $SP_i$ denotes a set of closed sequential patterns in document $d_i$ and $|p|$ indicates the length of pattern $p$. The extracted low-level terms are used to score a test document based on the total weight of the terms contained in the document.

- **IPE** [112]: The data mining method was proposed to refine the quality of discovered knowledge (features) obtained by closed sequential pattern mining. The main idea of IPE is to make use of mining non-relevant information (features) from some negative documents in the training dataset to remove or update the (deployed) relevant patterns extracted by the deploying method [99].

For all the above data mining methods, we set the minimum support threshold ($min\_sup$) to 0.20 (20% of the number of paragraphs in a document) since this value was recommended by these studies.

The rest of this section is dedicated for the the methods of mining discriminative patterns used for evaluation.

- **EPMine** [23]: This methods discovers discriminative patterns called *emerging patterns*, in a training set of examples. Coupled with the growth rate (GR) measure, an emerging pattern is defined as a frequent pattern whose the GR support is no less than a minimum threshold $\rho$, where $\rho \geq 1$. In this

context, the GR is defined as the ration of the pattern's (relative)support in the classes of relevant documents $D^+$ and non-relevant documents $D^-$.

Based on the above definition, the GR of a given pattern $\alpha$ can be calculated as follows:

$$GR(\alpha) = \frac{|\{d_i | d_i \in D^+, \alpha \subseteq d_i\}|}{|\{d_j | d_j \in D^-, \alpha \subseteq d_j\}|} \tag{7.5}$$

where $d_i$ and $d_j$ denote two documents in the training set $D = D^+ \cup D^-$. The discovery of emerging patterns have been shown to be useful for classification since it contains informative patterns which contrast two classes.

- **DPMine** [17]: This method focuses on finding highly relevant patterns in a class of interest with very low redundancy. It uses a sequential covering algorithm called *MMRFS* with information gain (IG) or Fisher score as a measure of relevance of a pattern. For each iteration, this algorithm selects the pattern with the highest score estimated by a gain function $g$. Given a set of already selected patterns $\Psi$, the gain of a given pattern $\alpha$ is

$$g(\alpha) = Rel(\alpha, c) - \max_{\beta \in \Psi} Red(\alpha, \beta) \tag{7.6}$$

where $Rel(\alpha, c)$ denotes the relevance of $\alpha$ w.r.t. the class of interest $c$ and $Red(\alpha, \beta)$ means the redundancy between two patterns $\alpha$ and $\beta$. Based on the definition of gain function, $\alpha$ is selected if it is highly relevant to the class $c$ and contains very low redundancy with the already selected patterns $\beta \in \Psi$.

In this experiment, *Rel* is modelled by IG measure (see Chapter 2) since it has been successfully applied to text categorization [83, 107]. Given a training set $D = D^+ \cup D^-$, the redundancy of two patterns $\alpha$ and $\beta$ is measured by a variant of Jaccard measure [17] as follows:

$$Red(\alpha, \beta) = \frac{cover(\alpha,\beta)}{cover(\alpha)+cover(\beta)-cover(\alpha,\beta)} \times min(Rel(\alpha), Rel(\beta)) \quad (7.7)$$

where $cover(\alpha, \beta) = \{d | d \in D, \alpha \subseteq d \Rightarrow \beta \subseteq d\}$ and $Rel(\alpha)$ denotes the relevance of $\alpha$.

The advantage of DPMine is that the number of selected features is automatically determined by the coverage constraint $\delta$. This parameter is set to ensure that each training instance (e.g., document) is covered at least $\delta$ times by the selected features.

### 7.4.2 Term-based approaches

The second category of our baseline methods includes the popular term-based methods for relevance feedback:

- **Rocchio** [46]: This method generates a Centroid for representing user profiles by extracting terms from positive documents and performing to revise weights of the terms with negative documents. The centroid $\vec{c}$ of a topic can be generated as follows:

$$\vec{c} = \alpha \frac{1}{|D^+|} \sum_{\vec{d} \in D^+} \frac{\vec{d}}{||\vec{d}||} - \beta \frac{1}{|D^-|} \sum_{\vec{d} \in D^-} \frac{\vec{d}}{||\vec{d}||} \quad (7.8)$$

where $||\overrightarrow{d}||$ be normalized vector for document $d$. $\alpha$ and $\beta$ be a control parameter for the effect of relevant and non-relevant data respectively. According to [14, 46], there are two recommendations for setting the two parameters: $\alpha = 16$ and $\beta = 4$; and $\alpha = \beta = 1.0$. We have tested both accommodations on assessor topics and found the latter recommendation was the best one. Therefore, we let $\alpha = \beta = 1.0$.

- **BM25** [75] is one of state-of-the-art term-based models. The term weights are estimated using the following probabilistic model-based equation:

$$W(t) = \frac{tf \cdot (k_1 + 1)}{k_1 \cdot ((1-b) + b\frac{DL}{AVDL}) + tf} \cdot \log \frac{\frac{(r+0.5)}{(n-r+0.5)}}{\frac{(R-r+0.5)}{(N-n-R+r+0.5)}} \tag{7.9}$$

where $N$ is the total number of documents in the training set; $R$ is the number of positive documents in the training set; $n$ is the number of documents which contain term $t$; $r$ is the number of positive documents which contain term $t$; $tf$ is the term frequency; $DL$ and $AVDL$ are the document length and average document length, respectively; and $k_1$ and $b$ are the experimental parameters, where the recommended values of $k_1$ and $b$ for this data collection are 1.2 and 0.75, respectively [112].

- **Ranked-based Support Vector Machine (SVM)**: The linear SVM has been demonstrated useful for text classification and filtering [83]. Thus, we compare it with other baseline approaches. However, most existing SVMs were developed for binary decision rather than ranking documents. We choose the linear SVM modified for ranking documents in [112]. This SVM

only uses term-based features extracted from a set of training documents. We describe the details as following:

There are two classes: $y_i \in \{-1, 1\}$ where $+1$ is assigned to a document if it is relevant; otherwise it is assigned with $-1$ and there are $N$ labelled training examples: $(d_i, y_i), \ldots, (d_N, y_N)$, $d_i \in \mathbb{R}^n$ where $n$ is the dimensionality of the vector. Given a function $h(d) =< w \cdot d > +b$ where $b$ is the bias, $h(d) = +1$ if $< w.d > +b \geq 0$; otherwise $h(d) = -1$, and $< w \cdot d >$ is the dot product of an optimal weight vector $w$ and the document vector $d$. To find the optimal weight vector $w$ for the training set, we perform the following function: $w = \sum_{i=1}^{N} y_i \alpha_i d_i$ subject to $\sum_{i=1}^{l} \alpha_i y_i = 0$ and $\alpha_i \geq 0$, where $\alpha_i$ is the weight of the sample $d_i$. For the purpose of ranking, $b$ can be ignored and all training documents are important equally. We hence assign the same $\alpha_i$ value (i.e., 1) to each positive document first, and then determine the same $\alpha_i$ (i.e., $\alpha'$) value to each negative document. Thus, the optimal weight vector $w$ can be determined as following function:

$$w = \left( \sum_{d_i \in D^+} d_i \right) + \left( \sum_{d_j \in D^-} d_j \alpha' \right) \tag{7.10}$$

In order to score a matching document $d$, the ranking function $S(d) = d \cdot w$ is performed. A high positive value assigned to the document $d$ can imply that the document tends to be *highly relevant*.

For each RCV1 assessor topic, we choose top-150 terms in a set of relevant documents, based on $TF - IDF$ values for all the term-based models.

## 7.5 Parameter Setting

*PCMine* needs two parameters as input: (1) the minimum support threshold *min_sup* and (2) the number of $k$ offenders from non-relevant documents. The *min_sup* parameter was given as main input. The parameter $k$ was tuned for the RCV1 assessor datasets.

Beside the main parameter *min_sup*, *EPMine* and *DPMine* are also dependent on the specific parameters: the growth rate threshold $\rho$ and the sequential coverage threshold $\delta$ respectively. Unfortunately, both *DPMine* and *EPMine* do not provide any guideline to set the parameters. Thus, we need to empirically find the best values for these constraints.

To find the best value for *min_sup*, we run *PTMine* algorithm to discover (closed) sequential patterns for each RCV1 dataset. Without pattern cleaning, weights were computed and attached to each (closed)pattern based on Eq.(6.1) for use in the testing phase Figure 7.3 illustrates the MAP performance associated with different minimum support values. As seen in this figure, the best MAP performance of both closed sequential pattern and sequential pattern models is achieved at the minimum support 0.02 (2% of number of transactions contained in a document database), where the model with closed sequential patterns perform over the model with sequential ones though all the threshold values. Increasing the threshold values results in degrading the MAP performance. Figure 7.7 illustrates the numbers of (closed)sequential patterns discovered corresponding to different the minimum support values. According to Figure 7.7, the discovery of closed

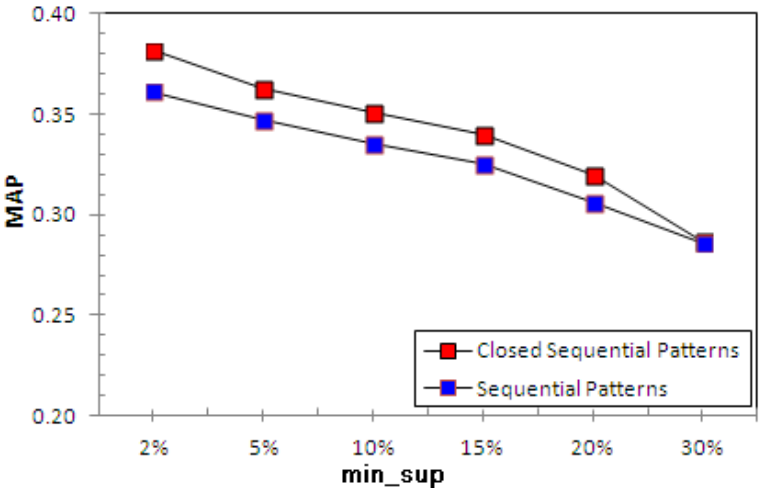



FIGURE 7.5: The MAP performance on the 50 assessor topics w.r.t. different *min_sup* values

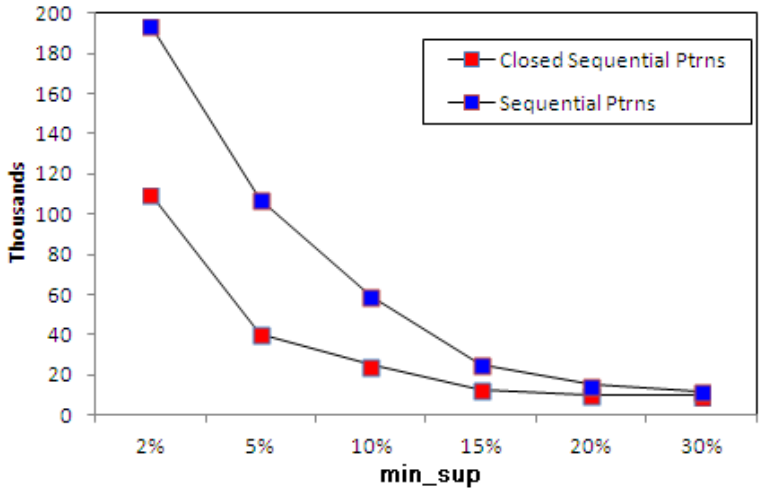



FIGURE 7.6: The comparison number of closed sequential and sequential patterns w.r.t. different *min_sup*

sequential patterns largely reduce the number of sequential patterns in the training datasets, especially at very low support. The results highlight the clear merit of mining closed sequential patterns in text since some noisy patterns and redundant patterns are removed. Based on these results, we fixed $min\_sup = 0.02$ for

parameter evaluation.

Here, we examine the specific parameter in *PCMine* algorithm which is the top$-k$ offenders. We can expect that a high value $k$ may result in retrieval of non-relevant documents which are not interesting and may increase noise to the learning model. On the other hand, a small value may miss useful non-relevant information used for identifying ambiguous patterns and describing the user's interest topic. Figure 7.8 illustrates the comparison performance of varying the $k$ parameter according to the number of relevant documents given in the training set.

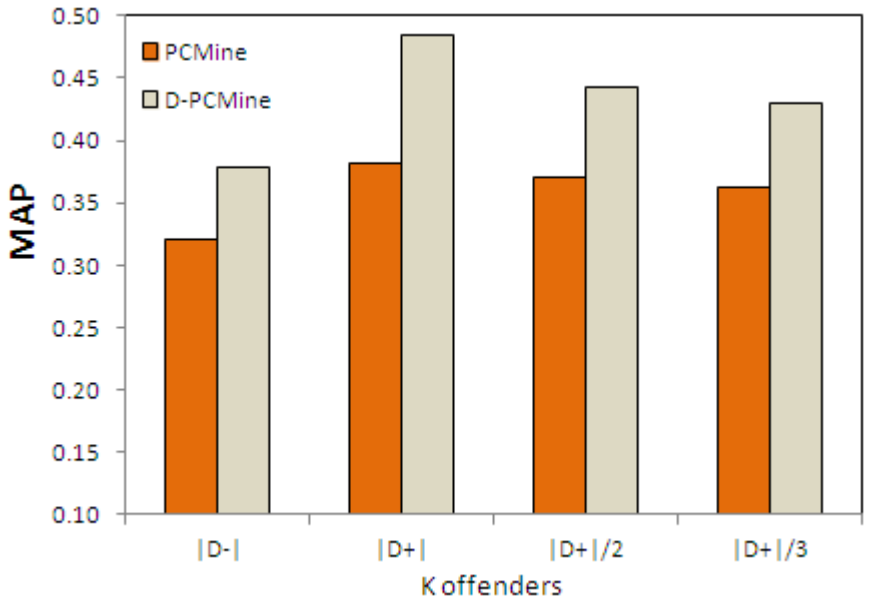



FIGURE 7.7: The MAP performance with different $k$ offenders at $min\_sup = 0.02$

According to this figure, the best MAP performance of *PCMine* is achieved when the number of $k$ offenders equals the number of relevant documents in the training dataset ($k = |D^+|$). In contrast, increasing or decreasing the number of

$k$ offenders tends to degrade the MAP score. For example, when considering all non-relevant documents as the offenders ($k = |D^-|$), the lowest performance is made comparing with the other values. As seen in Figure 7.8, *D-PCMine* also reaches the peak performance using the number of offenders which equals to the number of relevant documents. Thus, we fixed $k = |D^+|$.

By fixed $min\_sup = 0.02$, we tried to pick the best values for parameters $\delta$ and $\rho$ in *DPMine* and *EPMine* respectively. We apply these methods to select closed sequential patterns highly relevant to the topic of user's interest. Then, we use their result-set of relevant patterns to construct profile models in the same way as *PCMine*. Finally, we found that *DPMine* performs the best performance on the MAP score in the RCV1 datasets by setting $\delta = 20$. On the other hand, we set $\rho = 3.0$ for *EPMine* with the same reason.

## 7.6 Experiments

We conducted three experiments using the RCV1 dataset. The purpose of the first experiment is to confirm the intuitiveness of the relevant patterns obtained by *PCMine*. The second experiment is aimed to demonstrate the usefulness of relevant patterns to improve the effectiveness of information filtering. In the last one, we demonstrate the computation performance of *PTMine* for mining relevant patterns on the search space of (closed)sequential patterns.

### 7.6.1 Evaluation of relevant patterns

In this section, we compare the effectiveness of relevant patterns obtained by our proposed method, *PCMine*, with the two baseline methods, including *DPMine* and *EPMine*.

To make a comparison among these methods, we first get all closed sequential patterns for each RCV1 topic using PTMining. Then, we apply these methods to extract those patterns relevant to the class of user's interest (i.e., $D^+$). We set the coverage parameter $\delta = 20$ for *DPMine* and the growth rate threshold $\rho = 3.0$ for *EPMine* to select relevant patterns. This parameter setting has been found to result in these methods performed their best in this data collection (see Section 7.5). Finally, filtering models were built using the relevant patterns obtained by these methods. Table 7.6 compares the effectiveness of the two baselines to the proposed *PCMine* method. As seen in this table, *PCMine* is always more

| **Method** | **top-20** | **MAP** | $b/p$ | $F_{\beta=1}$ | **#Ptrns. topic** |
|---|---|---|---|---|---|
| PCMine | 0.451 | 0.382 | 0.381 | 0.400 | 168.88 |
| DPMine | 0.445 | 0.378 | 0.377 | 0.393 | 97.78 |
| EPMine | 0.443 | 0.375 | 0.371 | 0.388 | 154.88 |
| PTMining (Cls. Seq. Ptrns) | 0.355 | 0.312 | 0.31 | 0.341 | 585.88 |

TABLE 7.5: Comparison of relevant patterns by *PCMine* against the other baselines on the first 50 RCV1 topics

effective than any of the two baselines with the effectiveness measures. Compared to *PTMining* which uses solely closed sequential patterns, both *DPMine* and *EPMine* achieve significantly better performance on all the metrics with fewer patterns (i.e., 98.78 patterns by *DPMine* and 154.88 patterns by *EPMine*). This

result highlights the importance of selecting patterns highly relevant to the class of interest.

Compared to *EPMine* which uses the top closed patterns with a growth rate threshold, *DPMine* achieves better performance on all the metrics. This can be described by the reason that the quality of relevant patterns obtained by *EPMine* rely on how a good value for the threshold is obtained, which is not obvious. In contrast to *EPMine*, the top closed patterns by *DPMine* is less sensitive to diverse characteristics of the RCV1 datasets and is easier for human users to specify.

Compared to the baselines, *PCMine* is the best performing method for discovering relevant patterns. There are two likely reasons for the improvements. The first reason is that the baselines mainly focus on selecting the most relevant patterns with a measure of relevance of a pattern. As a result, moderately relevant patterns may be easily missed with inappropriate threshold value. In contrast to the baselines, *PCMine* have benefits from large informative patterns by carefully eliminating ambiguous ones. The second reason is the robustness. The relevant patterns obtained by the baselines are solely selected from a large pool of positive patterns in relevant documents. However, in many cases the number of relevant documents is rare and may be hardly representative of all relevant documents in the discussion topic due to expensive human labelling. These topics are known as *bias* [103]. As a consequence, the quality of relevant features extracted from the limited number of relevant documents will easily get hurt. Conversely, the relevant features obtained by *PCMine* are selected from both positive and negative patterns, leading to the robustness. Table 7.6 illustrates the precisions of the top-20 obtained by *PCMine* and the baselines on the first 10 RCV1 topics.

| **Topic** | $\|D^+\|$ | $\|D^-\|$ | **PTMining** | **DPMine** | **PCMine** | **EPMine** |
|---|---|---|---|---|---|---|
| 101 | 7 | 16 | 0.700 | 0.700 | **0.800** | 0.650 |
| 102 | 135 | 64 | 0.950 | 0.900 | 0.950 | **1.000** |
| 103 | 14 | 50 | 0.800 | 0.800 | **0.810** | 0.600 |
| 104 | 120 | 74 | 0.950 | **1.00** | 0.950 | 0.950 |
| 105 | 16 | 21 | 0.648 | **0.70** | 0.650 | 0.650 |
| 106 | 4 | 40 | 0.100 | 0.100 | 0.150 | **0.200** |
| 107 | 3 | 58 | 0.200 | 0.270 | **0.300** | 0.250 |
| 108 | 3 | 50 | 0.250 | 0.250 | **0.300** | 0.150 |
| 109 | 20 | 20 | 0.200 | **0.300** | 0.250 | 0.250 |
| 110 | 5 | 86 | 0.150 | 0.410 | **0.450** | 0.400 |
| AVG | 32.70 | 47.90 | 0.498 | 0.543 | **0.561** | 0.510 |

TABLE 7.6: Comparison PCM with data mining-based methods on precision of top 20 returned documents on 10 RCV1 topics

According to Table 7.7, the numbers of relevant documents and non-relevant documents in each topic are diverse. For example, topic 101 contains sever relevant documents out of twenty-three documents while topic 102 with 135 relevant documents out of 199 documents. As seen in this table, the performance of *PCMine* is more robust compared to the baselines especially in the topics with a few relevant documents such as topic 103, 107, 108, and 110). The improvement of *PCMine* is consistent and stable as shown in Figure 7.8.

### 7.6.2 Usefulness of relevant patterns

In the second experiment, we explore the usefulness of relevant patterns to improve the filtering performance. We focus on comparing its performance to that of existing approaches. We first compare our proposed approach to data mining-based methods that use closed sequential patterns. Then, we compare it to state-of-the-art term-based methods.

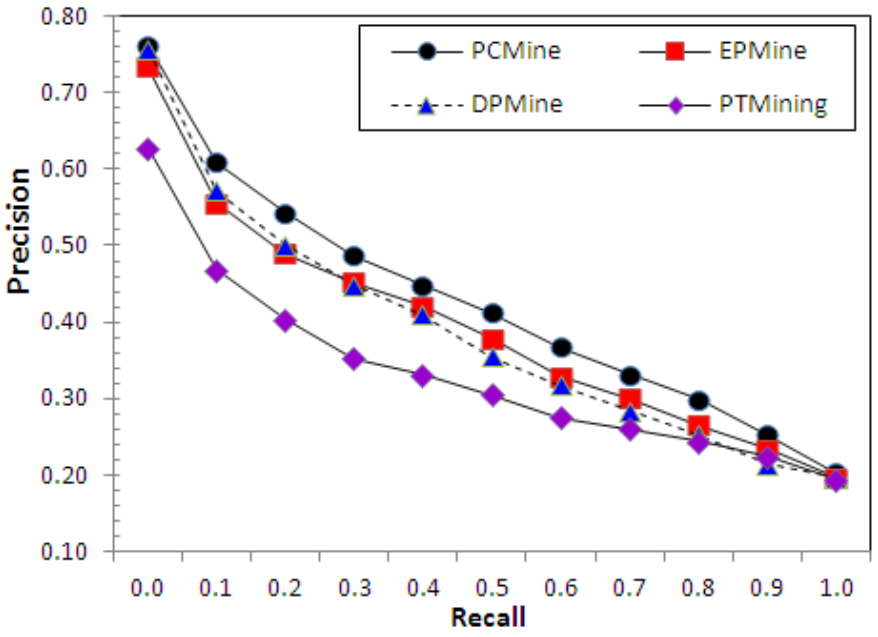



FIGURE 7.8: The Precision-Recall Curve

#### 7.6.2.1 Comparing against other patterns

According to the last section, we make use of the discovery of discriminative patterns using *DPMine* and *EPMine*. Then, we show that relevant patterns obtained by these methods could improve the significant performance comparing with closed sequential patterns.

To make a comprehensive evaluation, we implement two variants of pattern deployment that use the discriminative patterns for constructing low-level features to improve the filtering performance. The first baseline, denoted *D-DPMine*, is built by using the relevant patterns obtained by *DPMine* in training documents to evaluate the weights of low-level terms based on the pattern deploying approach (see Section 6.2.1) instead of closed sequential patterns. The second

baseline, denoted *D-EPMine*, also performs the construction of low-level features using the relevant patterns obtained by *EPMine*. Finally, the constructed (low-level)features are used to score test documents in the testing phase. Table 7.8 compares the performance of all the data mining-based baselines on the 50 assessor topics.

| **Method** | **#Feature per topic** | **top-20** | **MAP** | $b/p$ | $F_{\beta=1}$ |
|---|---|---|---|---|---|
| **D-PCMine** | 174.82(L) | **0.549** | **0.484** | **0.469** | **0.466** |
| D-DPMine | 46.04(L) | 0.492 | 0.453 | 0.432 | 0.445 |
| PDS [99] | 152.78(L) | 0.496 | 0.444 | 0.430 | 0.439 |
| IPE [112] | 100.02(L) | 0.493 | 0.444 | 0.430 | 0.429 |
| D-EPMine | 89.02(L) | 0.483 | 0.440 | 0.429 | 0.432 |
| **PCMine** | 168.88(H) | **0.451** | **0.382** | **0.381** | **0.400** |
| DPMine | 20.00(H) | 0.445 | 0.378 | 0.377 | 0.393 |
| EPMine | 98.44(H) | 0.443 | 0.375 | 0.371 | 0.388 |
| PTM [98] | 99.16(H) | 0.406 | 0.364 | 0.353 | 0.390 |

TABLE 7.7: Comparison of all data mining-based methods of the first 50 topics, where (L) means low-level terms and (H) means high-level patterns

According to this table, we can see that the proposed *PCMine* and *D-PCMine* methods are always more effective than the baselines across the metrics. The largest improvements are observed for top-20 and MAP. The results support the superiority of our proposed approach for constructing (low-level)features to improve the filtering performance.

The main observation from Table 7.8 is that most of the methods that make use of the discovery of relevant patterns outperform the ones that use closed sequential patterns (i.e., PTM, PDS, and IPE). Compared to PTM, both *EPMine* and *DPMine* achieve better overall performance with fewer patterns. This result highlights the positive effects of discriminative pattern mining for representing

relevant concepts in the user's interest topic. However, the encouraging performance of *PCMine* is caused by the selection of positive patterns and negative patterns.

Another interesting point that we found in this table is that all the models that use features constructed by pattern deployment perform much better than the ones that treat patterns as atomic features. This can be explained by the problem of specific long patterns with low frequency. In addition, Table 7.8 clearly demonstrates the effective use of patterns for both term weighting and term selection.

Let us compare the performance of the models that use low-level features constructed by pattern deploying. We first could not find a significant difference between PDS which uses features created by purely positive patterns and IPE which makes use of mining non-relevant information to review the weights of positive features. Compared to *PDS* and *IPE*, *D-DPMine* which uses features constructed from discriminative patterns perform better overall performance. In contrast to *D-DPMine*, *D-EPMine* perform the lowest performance. The likely reason is the difficulties in finding the right threshold of $\rho$ to prune noisy patterns in number enough to avoid the overfitting effect. Compared to *D-DPMine* and *D-EPMine*, the significant improvement of *D-PCMine* is always consistent and stable. The first reason is that both *D-DPMine* and *D-EPMine* cannot deal with noisy features constructed from patterns including some general terms. Such features increase the mistaken retrieval of non-relevant documents. In *D-PCMine*, we deal with this issue by suppressing their weight based on their appearance in weak negative patterns. Furthermore, we use negative features to reduce the

mistaken retrieval of non-relevant documents that close to the user search intent. The second reason is that discriminative pattern mining focuses on finding highly relevant patterns in the class of interest using a threshold, which may easily miss many moderately relevant patterns not satisfying the threshold. Figure 7.9 and 7.10 clearly demonstrate the encouraging performance of *PCMine* and *D-PCMine* comparing with the baselines on the plotting of precision-recall.

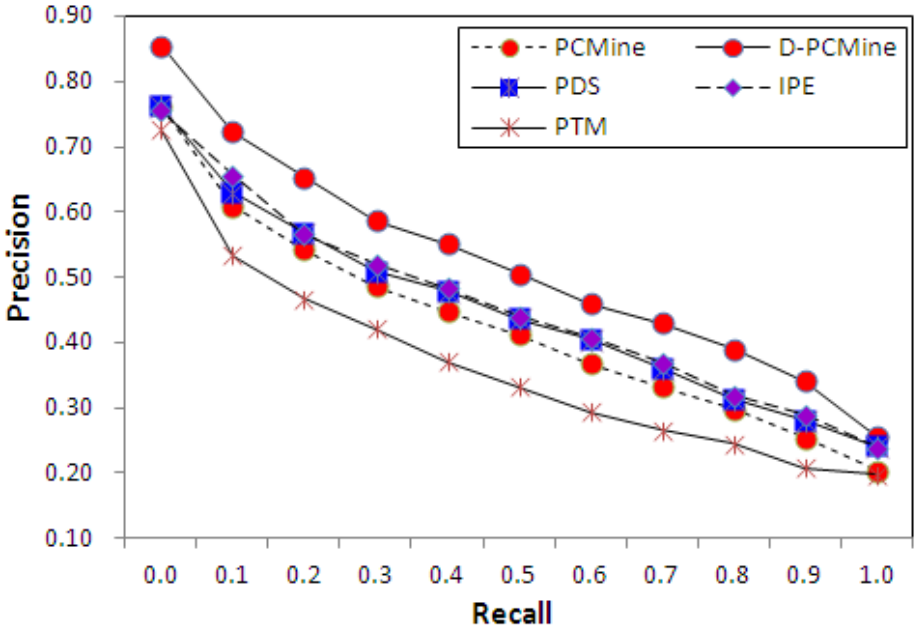



FIGURE 7.9: The Precision-Recall curve of *D-PCMine* and *PCMine* against the baselines that use closed sequential patterns

#### 7.6.2.2 Comparing against Classical IR Models

In this section, we compare the performance of *D-PCMine* to state-of-the-art term-based IR methods, including Rocchio, BM25, and SVM. As shown in Table 7.9, the substantially improvements are made on *D-PCMine* compared to all

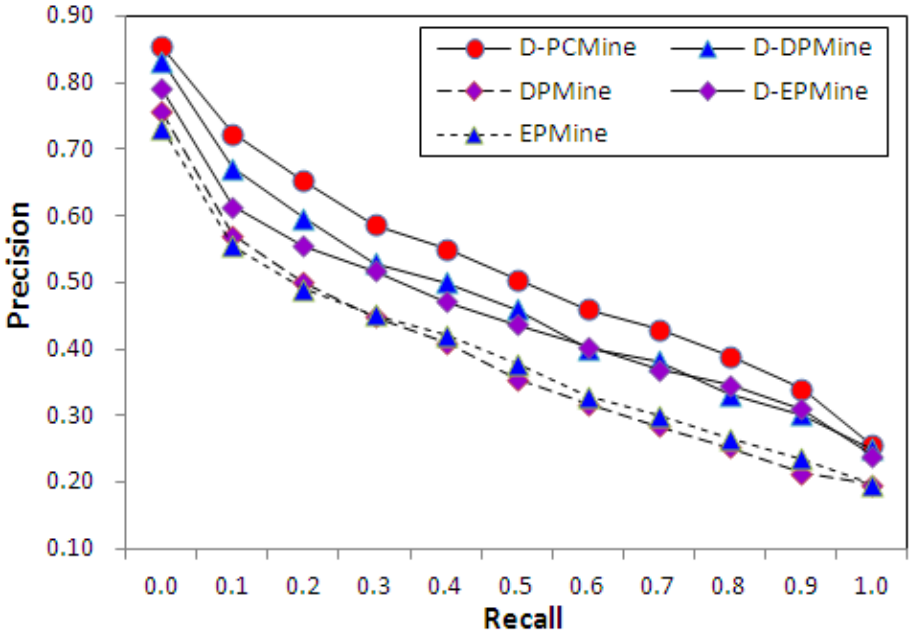



FIGURE 7.10: The Precision-Recall curve of *D-PCMine* against *EPMine* and *DPMine*

the term-based methods. The results demonstrate that extracting terms from frequent patterns which carry semantic relations among terms is much more effective than the ones extracted from raw documents since they are frequent and occur in specific contexts of these documents (i.e., sentences and paragraphs). The results also illustrate the effective use of patterns to improve the correctness of term weights without the assumption of term independence.

The plotting of precision-recall curve for *D-PCMine* and all the term-based methods on the assessor topics is illustrated in Figure 7.11.

Table 7.11 and 7.12 illustrates the t-test statistics of *PCMine* and *D-PCMine* respectively comparing with all the baseline models. This result confirm that their best improvement are statistically significant.

TABLE 7.8: Comparing the performance of PCM against state-of-the-art IR methods where %*chg* means the percentage change over the best term-based model

| Methods | *top-20* | *MAP* | *b/p* | $F_{\beta=1}$ |
|---|---|---|---|---|
| **D-PCMine** | **0.549** | **0.484** | **0.469** | **0.466** |
| Rocchio | 0.490 | 0.440 | 0.411 | 0.435 |
| BM25 | 0.469 | 0.418 | 0.420 | 0.387 |
| SVM | 0.447 | 0.408 | 0.409 | 0.421 |
| %*chg* | +12.04% | +10.00% | +14.11% | +7.12% |

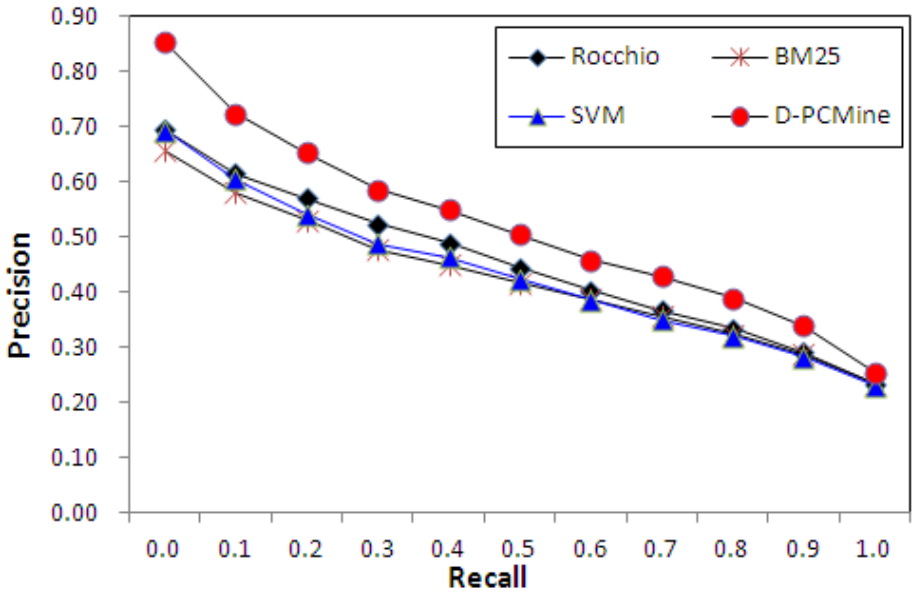



FIGURE 7.11: The Precision-Recall Curve *D-PCMine* and *PCMine* against term-based approaches

### 7.6.3 Computation Efficiency Evaluation

In this section, we investigate the computational performance of $PTMine$ algorithm for mining relevant patterns. Since the proposed $PTMine$ algorithm has been extended from $SPMine$ algorithm proposed in [98], we compare the

TABLE 7.9: $p-$values for the baseline methods comparing with *PCMine* in the assessor topics

| Model | *top-20* | *MAP* | *b/p* | $F_{\beta=1}$ |
|---|---|---|---|---|
| DPMine | 0.0012 | 0.0001 | 0.0065 | 0.0018 |
| EPMine | 0.0030 | 0.0026 | 0.0034 | 0.0006 |
| PTM | 0.0001 | 0.0005 | 0.0001 | 0.0002 |

TABLE 7.10: $p-$values for the baseline methods comparing with *D-PCMine* in the assessor topics

| Model | *top-20* | *MAP* | *b/p* | $F_{\beta=1}$ |
|---|---|---|---|---|
| D-DPMine | 0.0107 | 0.0002 | 0.0125 | 0.0001 |
| D-EPMine | 0.00242 | 0.0006 | 0.0132 | 0.0006 |
| IPE | 0.0272 | 0.0017 | 0.0065 | 0.0184 |
| PDS | 0.0215 | 0.0004 | 0.0249 | 0.0002 |
| Rocchio | 0.0048 | 0.0013 | 0.0264 | 0.0003 |
| BM25 | 0.0002 | 7.140E-5 | 0.007 | 6.853E-5 |
| SVM | 0.0001 | 5.575E-9 | 0.0002 | 4.196E-8 |

performance of $PTMine$ and $SPMine$ for mining relevant patterns by varying the minimum support threshold.

We run the experiments on all the assessor topics with 2,704 training documents (50,985 paragraphs in total). The reported running times are measured on a DELL machine with an Intel Core 2 Duo 3.0 GHz CPU and 3.21 GB of RAM. We vary *min_sup* from 2% to 30% of the number of transactions obtained by each topic. The experiment results is plotted in Figure 7.12. Generally, we observe that the runtime increases as we lower *min_sup*. Further, we can note that the runtime of $PTMine$ on the search space of closed sequential patterns is much lower than the runtime of $SPMine$, especially at the low levels of *min_sup*. This highlights the advantage of the integration of pattern taxonomy and the anti-monotone pruning, which can largely reduce the search space of finding the

relevant knowledge.

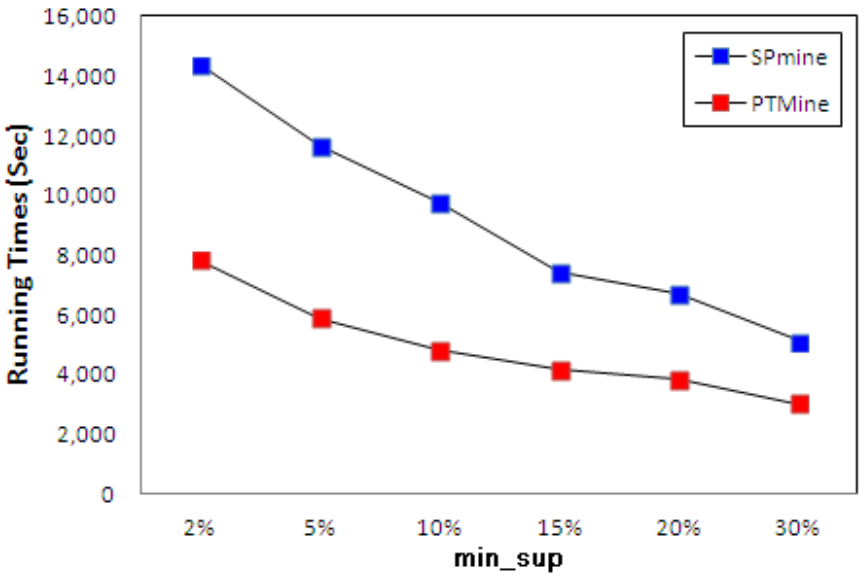



FIGURE 7.12: The running times of $PTMine$ and $SPMine$ on the close set of sequential patterns by varying the minimum support threshold $min\_sup$ with all the assessor topics, where $|D| = 2,704$ and $k = \frac{|D^+|}{2}$

We also plot the runtime of $PTMine$ and $SPMine$ for mining the relevant patterns on the search space of all sequential patterns as shown in Figure 7.13. As seen in this figure, we observe that $PTMine$ is still much more efficient than $SPMine$ for discovering the relevant patterns on the complete set of sequential patterns. Please note that $SPMine$ could not find the complete set of sequential patterns on some RCV1 topics at $min\_sup = 2\%$.

## 7.7 Discussion

In the previous section, we demonstrate the performance study of our proposed approach on information filtering. The experimental results have confirmed the

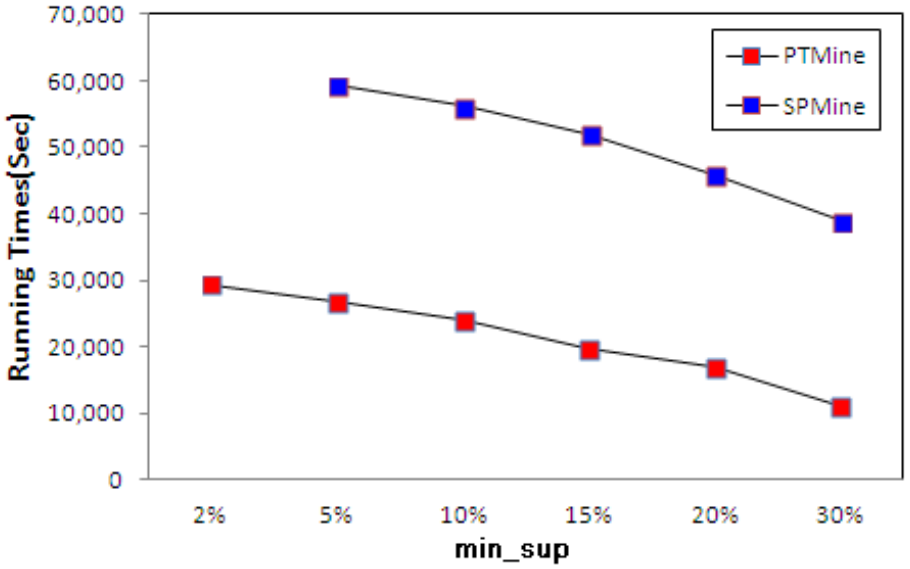



FIGURE 7.13: The running times of $PTMine$ and $SPMine$ on the complete set of sequential patterns by varying the minimum support threshold $min\_sup$ with all the assessor topics, where $|D| = 2,704$ and $k = \frac{|D^+|}{2}$

significant improvements made on the proposed approach in comparing with all baseline approaches. In this section, we discuss the issues of offender selection and the quality of relevant knowledge.

### 7.7.1 Offender Selection

Although relevant documents are much more important than non-relevant ones for describing user's information needs, the relevant information may be insufficient for relevant feature discovery since these relevant documents usually contain noises. We believe that non-relevant important is also important since this kind of information is helpful for identifying the non-relevant documents. Consequently,

| **K** | Avg. number of doc. | | | **Top-k** | **MAP** |
|---|---|---|---|---|---|
| | $D^+$ | $D^-$ | $D^-_{off}$ | | |
| 0 | 12.78 | 41.3 | 0.0 | 0.498 | 0.457 |
| $\frac{\lvert D^+\rvert}{2}$ | 12.78 | 41.3 | 6.06 | 0.547 | 0.486 |
| $\frac{\lvert D^+\rvert}{3}$ | 12.78 | 41.3 | 3.98 | 0.543 | 0.484 |
| $\lvert D^+\rvert$ | 12.78 | 41.3 | 10.26 | **0.549** | **0.487** |
| $\lvert D^-\rvert$ | 12.78 | 41.3 | 40.1 | 0.355 | 0.311 |

TABLE 7.11: The performance of $DPCMine$ with different top-$k$ offenders at $min\_sup = 0.02$

the combination of relevant and non-relevant information can help to improve the effectiveness of relevant feature discovery.

The challenging issue is how to find useful non-relevant documents from a large collection of non-relevant documents that may contain a lot of noisy information for describing a given topic. This is since these non-relevant documents can be easily collected from various topics. Table 7.11 illustrates the performance of setting of $k$ offenders for $DPMine$ on all the assessor topics.

As seen in this table, using all non-relevant documents as offenders ($k = |D^-|$) undermines the overall effectiveness as compared to using only relevant documents ($k = 0$). This may be explained by the reason that the noise offenders will weaken the importance of extracted relevant features. Conversely, selecting the non-relevant documents in numbers less than or equal the number of relevant documents generally results in better performance. The best performance is achieved if the number of $k$ offenders as equal as the number of relevant documents in the training set.

### 7.7.2 The quality of relevant knowledge

The obvious problem of using data mining for relevant feature discovery is that traditional data mining algorithms find patterns in numbers too large to be useful. Some of the discovered patterns are noise since many feedback datasets contain a lot of irrelevant and redundant information. Technically, selecting patterns relevant to the class of user's interest can reduce the issue of extracted noisy patterns and finally improve accuracy. Table 7.12 and 7.13 show the effects of using different groups of positive and negative patterns to describe the relevant knowledge (low-level features).

| **Group** | Top-20 | MAP | $F_{\beta=1}$ | Avg. no. of extracted ptrns. | |
|---|---|---|---|---|---|
| | | | | $\#RP^{+}$(CF) | $\#RP^{-}$(CF) |
| (1)Positive Ptrns. | 0.498 | 0.457 | 0.443 | 156.33(0%) | 0.0(−100%) |
| (2)Negative Ptrns. | 0.093 | 0.160 | 0.215 | 0.0(−100%) | 282.77(0%) |
| (3)**Relevant Ptrns** | **0.505** | **0.443** | **0.435** | **45.78**(−70%) | 0.0(−100%) |
| (4)Weak Pos. Ptrns. | 0.406 | 0.429 | 0.417 | 70.77(−55%) | 0.0(−100%) |
| (5)Weak Neg. Ptrns. | 0.093 | 0.160 | 0.218 | 0.0(−100%) | 52.43(−81%) |
| (6)Conflict Ptrns. | 0.341 | 0.338 | 0.351 | 39.87(−75%) | 0.0(−100%) |

TABLE 7.12: The results of using single groups of relevant patterns obtained by the proposed approach in all the assessor topics, where compression factor ($CF\% = (1 - \frac{|RP|}{|P|}) \times 100\%$) of original closed patterns ($P$) and extracted patterns ($RP$) ($min\_sup = 0.02$ and $K = \frac{|D^{+}|}{2}$)

The main observation from Table 7.12 is that positive patterns obtained from mining relevant documents are clearly more important than negative ones obtained from mining non-relevant documents to describe the relevant features. For example, the group of all positive closed patterns (1) perform better than the group of all negative patterns (2) or weak negative patterns (3). Table 7.12 and Figure 7.12 clearly illustrate the group of relevant patterns (3) outperform

over all the others with the compression ratio $70\% = (1 - \frac{45.78}{156.33}) \times 100\%$ of all closed sequential patterns in relevant documents. This can be explained by the fact that the group of relevant patterns contains clearly important information for describing the user relevance. The improvement of the relevant patterns in the precision-recall curve is also consistent.

Compared to relevant patterns (3), a great reduction of the performance made on the group of weak positive patterns (4) with 55% of the compression ratio is caused by the influence of some noisy (general) features extracted from these patterns. Such noise could not be removed using traditional pattern mining techniques since it is a part of pattern.

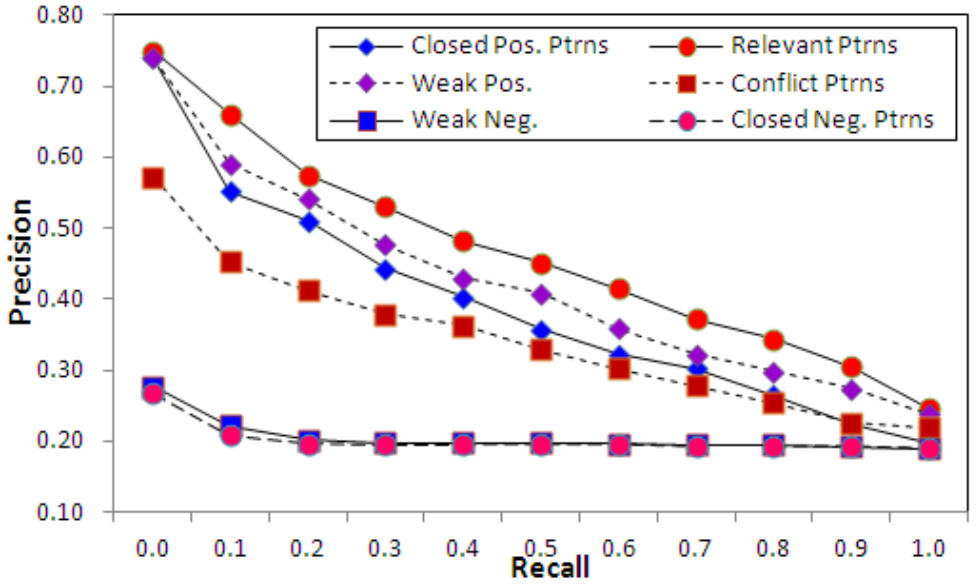



FIGURE 7.14: Comparison results of used different groups of patterns in RCV1 dataset

Table 7.13 compares the results of combining multiple groups of positive and negative patterns in Table 7.12. As seen in Table 7.13, the group of negative patterns

(2) will weaken the importance of positive patterns in each group. For example, the group (1)+(2) which uses all positive and all negative patterns results in the worst performance comparing with the other groups in this table. The degrading performance can be also made on the groups which use the negative patterns. The likely reason is that most of the negative patterns are noisy for a given topic due to the natural diversity of non-relevant documents.

Table 7.13 also illustrates the effective use of weak negative patterns (5) which removes many noisy patterns in the negative ones with 52.43 patterns (or 19% of the negative patterns) Compared to the group (3), the combination of relevant patterns and weak negative patterns ((3)+(5)) achieves better performance. From this table, the best performance was made on the combined group of relevant patterns, weak positive patterns, and weak negative ones ((3)+(4)+(5)). The use of weak negative patterns in this group is to suppress the extracted noisy features from weak positive patterns and to reduce the mistaken retrieval of non-relevant documents. Figure 7.13 clearly demonstrates the performance of the groups of relevant patterns on the precision-recall curve.

In summary, the experimental results support the effective strategy of using positive patterns and negative patterns in relevant feature discovery. We can conclude that although positive patterns are very significant for relevance feature discovery, some negative patterns are really useful to maintain the performance of relevance feature discovery and to reduce noise in relevance feedback.

| **Group** | Top-20 | MAP | $F_{\beta=1}$ | Avg. no. of extracted ptrns. | |
|---|---|---|---|---|---|
| | | | | $\#RP^{+}$(CF) | $\#RP^{-}$(CF) |
| (1)+(2) | 0.356 | 0.312 | 0.342 | 155.33(0%) | 282.77 (0%) |
| (2)+(3) | 0.424 | 0.377 | 0.369 | 47.48(−26%) | 282.77(0%) |
| (2)+(4) | 0.403 | 0.347 | 0.351 | 70.77(−55%) | 282.77(0%) |
| (3)+(4) | 0.499 | 0.432 | 0.430 | 116.45(−26%) | 0.0(−100%) |
| (3)+(5) | 0.537 | 0.482 | 0.461 | 47.48(−68%) | 52.43(−81%) |
| (4)+(5) | 0.516 | 0.434 | 0.443 | 70.77(−55%) | 52.43(−81%) |
| **(3)+(4)+(5)** | **0.549** | **0.485** | **0.466** | 116.45(−26%) | 52.43(−81%) |

TABLE 7.13: The results of combining multiple groups of relevant patterns in Table 7.12, where (1) = positive patterns, (2) = negative patterns, (3) = relevant patterns, (4) = weak positive patterns, and (5) = weak negative patterns

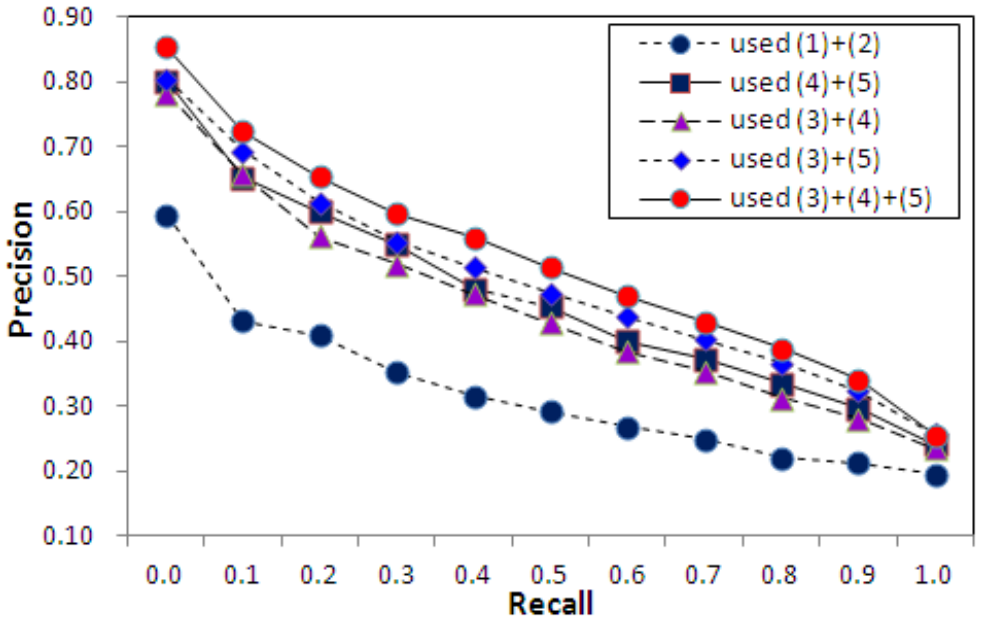



FIGURE 7.15: Comparison results of used combined groups of relevant patterns in RCV1 dataset, where (1) = positive patterns, (2) = negative patterns, (3) = relevant patterns, (4) = weak positive patterns, and (5) = weak negative patterns

# Chapter 8

# Conclusion and Future Work

Mining useful features to help users searching for relevant information is a challenging task in information retrieval and data mining. User relevance feedback is the most valuable source of information to acquire information needs of individual users. However, too much noise available in real-world feedback data can adversely affect the quality of extracted features.

The major research issue in this thesis is how to extract useful knowledge in user relevance feedback to reduce the effect of noisy features extracted by frequent pattern mining. This thesis presents a new pattern-based approach to relevance feature discovery. We introduce the concept of pattern cleaning, refining the quality of discovered frequent patterns in relevant documents using the selected non-relevant samples. We show that the information from the non-relevant samples is very useful to reduce noisy information in relevant documents as well as improve the quality of specific features to retrieve accurate information.

Numerous experiments within information filtering domain have been conducted

in this thesis. The latest version of the Reuters dataset, RCV1, is selected and tested by the proposed approaches to information filtering. The results illustrate that the proposed method outperform over several pure data mining-based methods as well as classical term-based methods in information filtering and text mining.

The main contributions of this research and the future work can be listed as:

- **Efficient Relevant Feature Mining**: relevant feature mining often involves searching a large space of features. Although efficient algorithms for frequent pattern mining are available, so far there has been very little attention focused to perform efficiently mining relevant patterns, excepting for [18], which mines directly relevant patterns using divide-and-conquer strategy.

  To the best our knowledge, the proposed method of pattern cleaning is the first anti-monotone pruning algorithm for relevant feature mining in a training dataset. It allows to prune efficiently ambiguous patterns with anti-monotone property. However, the proposed technique is *post-processing*, which requires the complete sets of positive and negative patterns as input.

  Recently, the focus was more on exploring new constraints for efficiently mining interesting patterns [20, 21, 34]. Such interestingness constraints could be pushed into the mining process to improve the efficiency. As the anti-monotone pruning, it would be best if a direct mining approach could be performed to remove early conflict patterns and non-relevant patterns without generating the complete sets of patterns.

- **New strategies for using negative patterns**

  In this thesis, we demonstrate that relevant feature discovery has benefits from the discovery of positive and negative patterns. Negative patterns can be used not only to filter out ambiguous patterns, but also to describe the precise knowledge. Instead of all negative patterns, the discovery of weak negative patterns from non-relevant offenders has shown to be useful for improving the effectiveness of relevant feature discovery.

  We believe that the use of negative patterns is useful and important for relevant feature discovery. Thus, it is worth to explore new strategies for using negative patterns to further improve the performance.

- **Exploring tasks of relevant feature discovery**

  Our experimental results have demonstrated that the proposed approach results in the encouraging improvements on the performance of information filtering. Thus, we would apply our proposed approach to text mining applications. For example, text categorization which is closely related to information filtering [83].

  Another direction can be made on exploring other application domains. Data mining have been applied to discover knowledge of user interest from various kinds of relevance feedback data such as image retrieval [61, 73], time series [49] and multimedia data [77]. Thus, it would be interesting to explore these kinds of feedback data.

# Appendix A

# An Example of an RCV1 Document

```
<?xml version="1.0" encoding="iso-8859-1" ?>
<newsitem itemid="2330" id="root" date="1996-08-20" xml:lang="en">
<title>USA: Tylan stock jumps; weighs sale of company.</title>
<headline>Tylan stock jumps; weighs sale of company.</headline>
<dateline>SAN DIEGO</dateline>
<text>
<p>The stock of Tylan General Inc. jumped Tuesday after the maker of
process-management equipment said it is exploring the sale of the
company and added that it has already received some inquiries from
potential buyers.</p>
<p>Tylan was up $2.50 to $12.75 in early trading on the Nasdaq market.</p>
<p>The company said it has set up a committee of directors to oversee
the sale and that Goldman, Sachs &amp;amp; Co. has been retained as its
financial adviser.</p>
</text>
<copyright>(c) Reuters Limited 1996</copyright>
<metadata>
<codes class="bip:countries:1.0">
  <code code="USA"> </code>
</codes>
<codes class="bip:industries:1.0">
  <code code="I34420"> </code>
</codes>
<codes class="bip:topics:1.0">
  <code code="C15">  </code>
  <code code="C152"> </code>
  <code code="C18">  </code>
  <code code="C181"> </code>
  <code code="CCAT"> </code>
</codes>
<dc element="dc.publisher" value="Reuters Holdings Plc"/>
<dc element="dc.date.published" value="1996-08-20"/>
<dc element="dc.source" value="Reuters"/>
<dc element="dc.creator.location" value="SAN DIEGO"/>
<dc element="dc.creator.location.country.name" value="USA"/>
<dc element="dc.source" value="Reuters"/>
</metadata>
</newsitem>
```

FIGURE A.1: An example of Reuters Corpus Volume 1 document

# Appendix B

# The Results in RCV1 50 topics

| Topic | \|Training\| | \|Training+\| | \|Testing\| | \|Testing+\| | Top20 | MAP | B/P | F1 | IAP | 0.0 | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 | 1.0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 101 | 23 | 7 | 577 | 307 | 1.00000 | 0.87031 | 0.80782 | 0.62670 | 0.87449 | 1.00000 | 1.00000 | 0.97438 | 0.95000 | 0.93893 | 0.91477 | 0.88517 | 0.86490 | 0.80128 | 0.76584 | 0.52414 |
| 102 | 199 | 135 | 308 | 159 | 1.00000 | 0.86281 | 0.87987 | 0.60200 | 0.86059 | 1.00000 | 1.00000 | 0.98000 | 0.98000 | 0.92045 | 0.92045 | 0.87273 | 0.82353 | 0.74576 | 0.71359 | 0.51002 |
| 103 | 64 | 14 | 528 | 61 | 0.71000 | 0.63432 | 0.54098 | 0.61401 | 0.54125 | 0.98000 | 0.72222 | 0.54386 | 0.54386 | 0.54386 | 0.54386 | 0.47664 | 0.47664 | 0.47664 | 0.46218 | 0.18399 |
| 104 | 194 | 120 | 279 | 94 | 0.85000 | 0.65560 | 0.51489 | 0.58879 | 0.71685 | 1.00000 | 1.00000 | 0.90909 | 0.90909 | 0.80000 | 0.80000 | 0.73077 | 0.50758 | 0.46429 | 0.39738 | 0.36719 |
| 105 | 37 | 16 | 258 | 50 | 0.90000 | 0.78616 | 0.66000 | 0.61033 | 0.72050 | 1.00000 | 1.00000 | 0.90909 | 0.90909 | 0.90909 | 0.69444 | 0.54545 | 0.52778 | 0.51899 | 0.50505 | 0.40652 |
| 106 | 44 | 4 | 321 | 31 | 0.10000 | 0.11083 | 0.11774 | 0.25232 | 0.15117 | 0.21429 | 0.14706 | 0.14706 | 0.14706 | 0.14706 | 0.14706 | 0.14706 | 0.14706 | 0.14706 | 0.14706 | 0.12500 |
| 107 | 61 | 3 | 571 | 37 | 0.35000 | 0.17326 | 0.24324 | 0.25884 | 0.28874 | 1.00000 | 0.57143 | 0.32130 | 0.30769 | 0.28571 | 0.15385 | 0.14371 | 0.12322 | 0.10680 | 0.09114 | 0.07129 |
| 108 | 53 | 3 | 386 | 15 | 0.49200 | 0.34723 | 0.58733 | 0.42001 | 0.39915 | 0.86957 | 0.60000 | 0.60000 | 0.60000 | 0.60000 | 0.34783 | 0.27273 | 0.20370 | 0.14286 | 0.08187 | 0.07212 |
| 109 | 40 | 20 | 240 | 74 | 0.55000 | 0.41410 | 0.36486 | 0.46278 | 0.72405 | 1.00000 | 1.00000 | 1.00000 | 1.00000 | 1.00000 | 0.97297 | 0.92500 | 0.51724 | 0.47706 | 0.39241 | 0.37430 | 0.30559 |
| 110 | 91 | 5 | 491 | 31 | 0.30000 | 0.22092 | 0.19355 | 0.33974 | 0.67171 | 0.90000 | 0.79412 | 0.79412 | 0.79412 | 0.79412 | 0.79412 | 0.79412 | 0.79412 | 0.79412 | 0.79412 | 0.07216 | 0.06366 |
| 111 | 52 | 3 | 451 | 15 | 0.15000 | 0.19289 | 0.13333 | 0.32814 | 0.18618 | 0.66667 | 0.37500 | 0.37500 | 0.16667 | 0.14000 | 0.07921 | 0.06569 | 0.04846 | 0.04416 | 0.04416 | 0.04298 |
| 112 | 57 | 6 | 481 | 20 | 0.50000 | 0.43078 | 0.39000 | 0.47480 | 0.60371 | 0.85714 | 0.80000 | 0.71429 | 0.71429 | 0.71429 | 0.71429 | 0.60870 | 0.60870 | 0.30303 | 0.30303 | 0.30303 |
| 113 | 68 | 12 | 552 | 70 | 0.35000 | 0.35032 | 0.45714 | 0.45100 | 0.42721 | 1.00000 | 0.51724 | 0.50000 | 0.50000 | 0.50000 | 0.50000 | 0.39326 | 0.36975 | 0.34266 | 0.25328 | 0.19033 | 0.13283 |
| 114 | 25 | 5 | 361 | 62 | 0.55000 | 0.35377 | 0.38710 | 0.49761 | 0.39276 | 0.66667 | 0.70588 | 0.68421 | 0.47500 | 0.28037 | 0.27434 | 0.26573 | 0.24719 | 0.24031 | 0.24031 | 0.24031 |
| 115 | 46 | 3 | 357 | 63 | 0.80000 | 0.49739 | 0.36508 | 0.52500 | 0.48295 | 1.00000 | 1.00000 | 0.76471 | 0.41935 | 0.41935 | 0.35345 | 0.35345 | 0.32857 | 0.28804 | 0.21190 | 0.17367 |
| 116 | 46 | 16 | 298 | 87 | 0.75000 | 0.64702 | 0.62069 | 0.56708 | 0.75772 | 1.00000 | 0.88462 | 0.88462 | 0.86486 | 0.83333 | 0.75000 | 0.68750 | 0.64800 | 0.64800 | 0.64800 | 0.48603 |
| 117 | 13 | 3 | 297 | 32 | 0.95000 | 0.74011 | 0.65625 | 0.60757 | 0.67264 | 1.00000 | 1.00000 | 1.00000 | 1.00000 | 1.00000 | 1.00000 | 0.90000 | 0.50000 | 0.31944 | 0.28261 | 0.25217 | 0.14480 |
| 118 | 32 | 3 | 293 | 14 | 0.25000 | 0.07119 | 0.07143 | 0.12684 | 0.10566 | 0.22222 | 0.15385 | 0.09231 | 0.09231 | 0.09231 | 0.09231 | 0.09231 | 0.09231 | 0.09231 | 0.09231 | 0.07000 | 0.07000 |
| 119 | 26 | 4 | 271 | 40 | 0.55000 | 0.40687 | 0.40000 | 0.50459 | 0.47961 | 1.00000 | 0.70000 | 0.50000 | 0.46429 | 0.43478 | 0.43478 | 0.41429 | 0.41429 | 0.38554 | 0.31034 | 0.21739 |
| 120 | 54 | 9 | 415 | 158 | 0.90000 | 0.71824 | 0.64557 | 0.58732 | 0.61920 | 0.62857 | 0.66474 | 0.66474 | 0.66474 | 0.66474 | 0.66474 | 0.66474 | 0.66474 | 0.63682 | 0.50352 | 0.38911 |
| 121 | 81 | 14 | 597 | 84 | 0.80000 | 0.70454 | 0.70238 | 0.58995 | 0.68398 | 1.00000 | 1.00000 | 0.92308 | 0.84615 | 0.83721 | 0.80769 | 0.71831 | 0.54128 | 0.32407 | 0.30894 | 0.21705 |
| 122 | 70 | 15 | 393 | 51 | 0.85000 | 0.66456 | 0.78431 | 0.57962 | 0.85174 | 1.00000 | 1.00000 | 1.00000 | 0.90476 | 0.86047 | 0.86047 | 0.86047 | 0.84000 | 0.75410 | 0.42839 |
| 123 | 51 | 3 | 342 | 17 | 0.35000 | 0.34010 | 0.35294 | 0.40652 | 0.30125 | 1.00000 | 0.50000 | 0.50000 | 0.21053 | 0.21053 | 0.21053 | 0.21053 | 0.21053 | 0.09036 | 0.08743 | 0.08333 |
| 124 | 33 | 6 | 250 | 33 | 0.15000 | 0.21952 | 0.26061 | 0.19567 | 0.19402 | 0.38727 | 0.26667 | 0.18182 | 0.18182 | 0.16304 | 0.16201 | 0.16201 | 0.16201 | 0.16201 | 0.15625 | 0.14932 |
| 125 | 36 | 12 | 544 | 132 | 0.95000 | 0.60008 | 0.53788 | 0.54815 | 0.47371 | 0.92220 | 0.53333 | 0.48000 | 0.46707 | 0.46707 | 0.46707 | 0.45078 | 0.43662 | 0.35294 | 0.32603 | 0.30769 |
| 126 | 29 | 19 | 270 | 172 | 0.90000 | 0.91417 | 0.95535 | 0.65190 | 0.89789 | 1.00000 | 0.96296 | 0.94737 | 0.94737 | 0.94737 | 0.89796 | 0.86286 | 0.86286 | 0.86286 | 0.84699 | 0.73820 |
| 127 | 32 | 5 | 238 | 42 | 0.35000 | 0.48539 | 0.26190 | 0.43250 | 0.59250 | 1.00000 | 0.83333 | 0.66667 | 0.66667 | 0.63333 | 0.60526 | 0.59091 | 0.50000 | 0.45455 | 0.34545 | 0.22133 |
| 128 | 51 | 4 | 276 | 33 | 0.25000 | 0.30790 | 0.39394 | 0.38259 | 0.33662 | 0.57143 | 0.40741 | 0.40741 | 0.40741 | 0.34043 | 0.33333 | 0.31343 | 0.27586 | 0.25225 | 0.20530 | 0.18857 |
| 129 | 72 | 17 | 507 | 57 | 0.75000 | 0.51373 | 0.47368 | 0.51099 | 0.46644 | 1.00000 | 0.75000 | 0.59714 | 0.58065 | 0.53488 | 0.35354 | 0.35354 | 0.33333 | 0.25000 | 0.22128 | 0.15650 |
| 130 | 24 | 3 | 307 | 16 | 0.40500 | 0.40146 | 0.43750 | 0.48123 | 0.43583 | 1.00000 | 1.00000 | 0.55556 | 0.55556 | 0.38889 | 0.32000 | 0.22222 | 0.22222 | 0.21667 | 0.16484 | 0.14815 |
| 131 | 31 | 4 | 252 | 74 | 0.90000 | 0.77373 | 0.66216 | 0.67195 | 0.71254 | 1.00000 | 0.85714 | 0.85714 | 0.85714 | 0.85714 | 0.85417 | 0.83636 | 0.54737 | 0.42857 | 0.40341 | 0.33945 |
| 132 | 103 | 7 | 446 | 22 | 0.07500 | 0.08578 | 0.25455 | 0.16968 | 0.16556 | 0.36360 | 0.36364 | 0.30000 | 0.11475 | 0.10227 | 0.09950 | 0.09950 | 0.09950 | 0.09950 | 0.09950 | 0.09950 | 0.07943 |
| 133 | 47 | 5 | 380 | 28 | 0.60000 | 0.50427 | 0.42857 | 0.50511 | 0.51371 | 1.00000 | 1.00000 | 0.75000 | 0.75000 | 0.75000 | 0.52174 | 0.48276 | 0.40000 | 0.31343 | 0.21698 | 0.11066 | 0.10526 |
| 134 | 31 | 5 | 351 | 67 | 0.15000 | 0.21332 | 0.17910 | 0.29819 | 0.25865 | 0.26230 | 0.26613 | 0.26613 | 0.26613 | 0.26613 | 0.26613 | 0.26613 | 0.26613 | 0.26613 | 0.26613 | 0.26613 | 0.18766 |
| 135 | 29 | 14 | 501 | 337 | 0.85000 | 0.76404 | 0.80415 | 0.60562 | 0.86708 | 1.00000 | 0.95122 | 0.85981 | 0.85981 | 0.85981 | 0.85981 | 0.85981 | 0.85981 | 0.85981 | 0.85981 | 0.84196 | 0.72600 |
| 136 | 46 | 8 | 452 | 67 | 0.15000 | 0.33788 | 0.26866 | 0.31130 | 0.35446 | 0.39860 | 0.39130 | 0.39130 | 0.39130 | 0.39130 | 0.39130 | 0.39130 | 0.39130 | 0.39130 | 0.39130 | 0.19551 | 0.17448 |
| 137 | 50 | 3 | 325 | 9 | 0.35000 | 0.63867 | 0.50556 | 0.59422 | 0.49027 | 1.00000 | 1.00000 | 0.60000 | 0.60000 | 0.60000 | 0.44444 | 0.35294 | 0.35294 | 0.29630 | 0.29630 | 0.22500 | 0.22500 |
| 138 | 98 | 7 | 328 | 44 | 0.55000 | 0.41038 | 0.50000 | 0.45485 | 0.33568 | 0.57143 | 0.55556 | 0.45455 | 0.34043 | 0.32759 | 0.29985 | 0.29412 | 0.28947 | 0.20455 | 0.20101 | 0.15493 |
| 139 | 21 | 3 | 253 | 17 | 0.55000 | 0.62488 | 0.64706 | 0.57094 | 0.69465 | 1.00000 | 1.00000 | 0.87500 | 0.80000 | 0.80000 | 0.75000 | 0.73333 | 0.66667 | 0.48387 | 0.26667 | 0.26563 |
| 140 | 59 | 11 | 432 | 67 | 0.60000 | 0.39822 | 0.44776 | 0.51498 | 0.54926 | 1.00000 | 0.90667 | 0.83333 | 0.72414 | 0.64286 | 0.50000 | 0.44086 | 0.40870 | 0.21012 | 0.19253 | 0.18265 |
| 141 | 56 | 24 | 379 | 82 | 0.80000 | 0.66071 | 0.54878 | 0.57438 | 0.57803 | 1.00000 | 0.90000 | 0.68966 | 0.57778 | 0.51087 | 0.51087 | 0.49550 | 0.46528 | 0.46528 | 0.42775 | 0.31538 |
| 142 | 28 | 4 | 198 | 24 | 0.15000 | 0.22610 | 0.41667 | 0.39631 | 0.40121 | 1.00000 | 0.70000 | 0.53333 | 0.53333 | 0.53333 | 0.35714 | 0.30000 | 0.26316 | 0.23288 | 0.20408 | 0.15493 | 0.13450 |
| 143 | 52 | 4 | 417 | 23 | 0.25000 | 0.17780 | 0.21739 | 0.25954 | 0.12886 | 0.47553 | 0.13158 | 0.11667 | 0.11667 | 0.09836 | 0.09836 | 0.07664 | 0.07664 | 0.07664 | 0.07664 | 0.07664 | 0.07372 |
| 144 | 50 | 6 | 380 | 55 | 0.80000 | 0.63431 | 0.54545 | 0.56145 | 0.76003 | 1.00000 | 0.90000 | 0.88889 | 0.88889 | 0.88889 | 0.80556 | 0.80000 | 0.72727 | 0.60000 | 0.47619 | 0.38462 | 0.38462 |
| 145 | 95 | 5 | 488 | 27 | 0.15000 | 0.11456 | 0.07407 | 0.19042 | 0.08822 | 0.11111 | 0.12500 | 0.08550 | 0.08550 | 0.08550 | 0.08550 | 0.08550 | 0.08550 | 0.08550 | 0.06824 | 0.06750 | 0.06750 |
| 146 | 32 | 13 | 280 | 111 | 0.75000 | 0.62426 | 0.54054 | 0.54770 | 0.67866 | 1.00000 | 1.00000 | 0.89655 | 0.77273 | 0.66116 | 0.56000 | 0.55285 | 0.54268 | 0.54268 | 0.47982 | 0.45679 | 0.45679 |
| 147 | 62 | 6 | 380 | 34 | 0.45000 | 0.46340 | 0.39412 | 0.48024 | 0.54847 | 0.86456 | 0.71429 | 0.61111 | 0.61111 | 0.55814 | 0.55814 | 0.55814 | 0.55814 | 0.54268 | 0.38961 | 0.33333 | 0.27666 |
| 148 | 33 | 12 | 380 | 228 | 0.95000 | 0.92898 | 0.90351 | 0.42165 | 0.89876 | 1.00000 | 0.96667 | 0.96203 | 0.96203 | 0.90452 | 0.90452 | 0.90452 | 0.90452 | 0.89706 | 0.86364 | 0.61687 | 0.61687 |
| 149 | 26 | 5 | 449 | 57 | 0.10000 | 0.17194 | 0.15263 | 0.23055 | 0.19428 | 0.35555 | 0.28412 | 0.22891 | 0.20408 | 0.16892 | 0.15309 | 0.15309 | 0.15309 | 0.15309 | 0.15309 | 0.15152 | 0.13164 |
| 150 | 51 | 4 | 371 | 54 | 0.35000 | 0.26954 | 0.16667 | 0.31725 | 0.27861 | 1.00000 | 0.25926 | 0.24926 | 0.22078 | 0.19549 | 0.19549 | 0.19549 | 0.19549 | 0.19549 | 0.19549 | 0.19549 | 0.16243 |

FIGURE B.1: The performance results of the proposed approach on the RCV1 50 assessor topics

# Bibliography

[1] F. Afrati, A. Gionis, and H. Mannila. Approximating a collection of frequent sets. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 12–19. ACM, 2004.

[2] R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In *Proceedings of the 20th International Conference on Very Large Data Bases*, pages 487–499. Morgan Kaufmann Publishers Inc., 1994.

[3] R. Agrawal and R. Srikant. Mining sequential patterns. In *Data Engineering, 1995. Proceedings of the Eleventh International Conference on*, pages 3–14. IEEE, 1995.

[4] R. Agrawal, T. Imieliński, and A. Swami. Mining association rules between sets of items in large databases. *ACM SIGMOD Record*, 22(2):207–216, 1993.

[5] M. Al Hasan and M.J. Zaki. Output space sampling for graph patterns. *Proceedings of the VLDB Endowment*, 2(1):730–741, 2009.

[6] A. Algarni, Y. Li, Y. Xu, and R.Y.K. Lau. An effective model of using negative relevance feedback for information filtering. In *Proceeding of the 18th ACM conference on Information and knowledge management*, pages 1605–1608. ACM, 2009.

[7] J. Allan. Relevance feedback with too much data. In *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 337–343. ACM, 1995.

[8] Ricardo A. Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1999. ISBN 020139829X.

[9] K. Barker and N. Cornacchia. Using noun phrase heads to extract document keyphrases. *Advances in Artificial Intelligence*, pages 40–52, 2000.

[10] R.J. Bayardo Jr. Efficiently mining long patterns from databases. *ACM Sigmod Record*, 27(2):85–93, 1998.

[11] N.J. Belkin and W.B. Croft. Information filtering and information retrieval: two sides of the same coin? *Communications of the ACM*, 35(12):29–38, 1992.

[12] Sergey Brin, Rajeev Motwani, and Craig Silverstein. Beyond market baskets: generalizing association rules to correlations. In *ACM SIGMOID*, pages 265–276, 1997.

[13] C. Buckley and E.M. Voorhees. Evaluating evaluation measure stability. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 33–40. ACM, 2000.

[14] C. Buckley, G. Salton, and J. Allan. The effect of adding relevance information in a relevance feedback environment. In *ACM SIGIR 17th International Conf.*, pages 292–300, 1994.

[15] D. Burdick, M. Calimlim, J. Flannick, J. Gehrke, and T. Yiu. Mafia: A maximal frequent itemset algorithm. *Knowledge and Data Engineering, IEEE Transactions on*, 17(11):1490–1504, 2005.

[16] T. Calders and B. Goethals. Mining all non-derivable frequent itemsets. *Principles of Data Mining and Knowledge Discovery*, pages 1–42, 2002.

[17] H. Cheng, X. Yan, J. Han, and C.W. Hsu. Discriminative frequent pattern analysis for effective classification. In *Proc. of ICDE*, pages 716–725. Citeseer, 2007.

[18] H. Cheng, X. Yan, J. Han, and P.S. Yu. Direct discriminative pattern mining for effective classification. In *Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on*, pages 169–178. IEEE, 2008.

[19] Y. Chi, S. Nijssen, R.R. Muntz, and J.N. Kok. Frequent subtree mining-an overview. *Fundamenta Informaticae*, 66(1):161–198, 2005.

[20] B. Crémilleux and A. Soulet. Discovering knowledge from local patterns with global constraints. *Computational Science and Its Applications–ICCSA 2008*, pages 1242–1257, 2008.

[21] L. De Raedt and A. Zimmermann. Constraint-based pattern set mining. In *Proceedings of SIAM International Conference on Data Mining 2007*, pages 1–12, 2007.

[22] G. Dong and J. Li. Efficient mining of emerging patterns: Discovering trends and differences. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 43–52. ACM, 1999.

[23] G. Dong, X. Zhang, L. Wong, and J. Li. Caep: Classification by aggregating emerging patterns. In *Discovery Science*, pages 737–737. Springer, 1999.

[24] A. Doucet and H. Ahonen-Myka. Non-contiguous word sequences for information retrieval. In *Proceedings of the Workshop on Multiword Expressions: Integrating Processing*, pages 88–95. Association for Computational Linguistics, 2004.

[25] H. Drucker, B. Shahrary, and D.C. Gibbon. Relevance feedback using support vector machines. In *MACHINE LEARNING-INTERNATIONAL WORKSHOP THEN CONFERENCE-*, pages 122–129, 2001.

[26] D.A. Evans and C. Zhai. Noun-phrase analysis in unrestricted text for information retrieval. In *Proceedings of the 34th annual meeting on Association*

*for Computational Linguistics*, pages 17–24. Association for Computational Linguistics, 1996.

[27] G. Fang, W. Wang, B. Oatley, B. Van Ness, M. Steinbach, and V. Kumar. Characterizing discriminative patterns. *arXiv preprint arXiv:1102.4104*, 2011.

[28] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. From data mining to knowledge discovery in databases. *AI magazine*, 17(3):37, 1996.

[29] J. Feng, F. Xie, X. Hu, P. Li, J. Cao, and X. Wu. Keyword extraction based on sequential pattern mining. In *Proceedings of the Third International Conference on Internet Multimedia Computing and Service*, pages 34–38. ACM, 2011.

[30] W.J. Frawley, G. Piatetsky-Shapiro, and C.J. Matheus. Knowledge discovery in databases: An overview. *Ai Magazine*, 13(3):57, 1992.

[31] K.R. Gee. Using latent semantic indexing to filter spam. In *Proceedings of the 2003 ACM symposium on Applied computing*, pages 460–464. ACM, 2003.

[32] L. Geng and H.J. Hamilton. Interestingness measures for data mining: A survey. *ACM Computing Surveys (CSUR)*, 38(3), 2006.

[33] K. Gouda and M.J. Zaki. Genmax: An efficient algorithm for mining maximal frequent itemsets. *Data Mining and Knowledge Discovery*, 11(3): 223–242, 2005.

[34] T. Guns, S. Nijssen, and L. De Raedt. k-pattern set mining under constraints. *Knowledge and Data Engineering, IEEE Transactions on*, 25(99): 1–1, 2010.

[35] J. Han. Cpar: Classification based on predictive association rules. In *Proceedings of the Third SIAM International Conference on Data Mining*, volume 3, pages 331–335. Society for Industrial & Applied, 2003.

[36] J. Han, J. Pei, B. Mortazavi-Asl, Q. Chen, U. Dayal, and M.C. Hsu. Freespan: frequent pattern-projected sequential pattern mining. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 355–359. ACM, 2000.

[37] J. Han, J. Pei, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal, and MC Hsu. Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth. In *Proceedings of the 17th International Conference on Data Engineering*, pages 215–224. Citeseer, 2001.

[38] J. Han, J. Pei, Y. Yin, and R. Mao. Mining frequent patterns without candidate generation: A frequent-pattern tree approach. *Data mining and knowledge discovery*, 8(1):53–87, 2004.

[39] J. Han, H. Cheng, D. Xin, and X. Yan. Frequent pattern mining: current status and future directions. *Data Mining and Knowledge Discovery*, 15 (1):55–86, 2007.

[40] P.J. Hayes and S.P. Weinstein. Construe/tis: a system for content-based indexing of a database of news stories. In *Second Annual Conference on Innovative Applications of Artificial Intelligence*, volume 97, 1990.

[41] E. Hernández-Reyes, R. García-Hernández, J. Carrasco-Ochoa, and J. Martínez-Trinidad. Document clustering based on maximal frequent sequences. *Advances in Natural Language Processing*, pages 257–267, 2006.

[42] W. Hersh, C. Buckley, TJ Leone, and D. Hickam. Ohsumed: an interactive retrieval evaluation and new large test collection for research. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 192–201. Springer-Verlag New York, Inc., 1994.

[43] Xiaodi Huang and Jianming Yong. A new approach to multimedia information filtering based on its structure. In *Proceedings of the 2004 Mutimedia Eurographics Symposium*, pages 59–68. Association for Computing Machinery (ACM), 2004.

[44] S. Jaillet, A. Laurent, and M. Teisseire. Sequential patterns for text categorization. *Intelligent Data Analysis*, 10(3):199–214, 2006.

[45] M. Jiang, E. Jensen, S. Beitzel, and S. Argamon. Choosing the right bigrams for information retrieval. *Classification, Clustering, and Data Mining Applications*, pages 531–540, 2004.

[46] T. Joachims. A probabilistic analysis of the rocchio algorithm with tfidf for text categorization. In *14th ICML International Conf. on Machine Learning*, pages 143–151, 1997.

[47] H. Joho and M. Sanderson. Document frequency and term specificity. In *Large Scale Semantic Access to Content (Text, Image, Video, and Sound)*, pages 350–359. LE CENTRE DE HAUTES ETUDES INTERNATIONALES D'INFORMATIQUE DOCUMENTAIRE, 2007.

[48] Yoshitaka Kameya and Taisuke Sato. Rp-growth: Top-k mining of relevant patterns with minimum support raising. In *Proceedings of the Twelfth SIAM International Conference on Data Mining, Anaheim*, pages 816–827. SIAM / Omnipress, 2012.

[49] E.J. Keogh and M.J. Pazzani. Relevance feedback retrieval of time series data. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 183–190. ACM, 1999.

[50] A. Knobbe and E. Ho. Pattern teams. *Knowledge Discovery in Databases: PKDD 2006*, pages 577–584, 2006.

[51] A.M. Lam-Adesina and G.J.F. Jones. Applying summarization techniques for term selection in relevance feedback. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 1–9. ACM, 2001.

[52] K. Lang. Newsweeder: Learning to filter netnews. In *In Proceedings of the Twelfth International Conference on Machine Learning*. Citeseer, 1995.

[53] V. Lavrenko and W.B. Croft. Relevance based language models. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 120–127. ACM, 2001.

[54] Y. Ledeneva, A. Gelbukh, and R. García-Hernández. Terms derived from frequent sequences for extractive text summarization. *Computational Linguistics and Intelligent Text Processing*, pages 593–604, 2008.

[55] D.D. Lewis, Y. Yang, T.G. Rose, and F. Li. Rcv1: A new benchmark collection for text categorization research. *The Journal of Machine Learning Research*, 5:361–397, 2004.

[56] J. Li, G. Dong, and K. Ramamohanarao. Making use of the most expressive jumping emerging patterns for classification. *Knowledge and Information systems*, 3(2):131–145, 2001.

[57] W. Li, J. Han, and J. Pei. Cmar: Accurate and efficient classification based on multiple class-association rules. In *Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on*, pages 369–376. IEEE, 2001.

[58] Y. Li, X. Zhou, P. Bruza, Y. Xu, and R.Y.K. Lau. A two-stage text mining model for information filtering. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 1023–1032. ACM, 2008.

[59] Y. Li, A. Algarni, S.T. Wu, and Y. Xue. Mining negative relevance feedback for information filtering. In *Web Intelligence and Intelligent Agent Technologies, 2009. WI-IAT'09. IEEE/WIC/ACM International Joint Conferences on*, volume 1, pages 606–613. IEEE, 2009.

[60] Y. Li, A. Algarni, and N. Zhong. Mining positive and negative patterns for relevance feature discovery. In *16th ACM SIGKDD International Conf. on Knowledge discovery and data mining*, pages 753–762, 2010.

[61] S.D. MacArthur, C.E. Brodley, and C.R. Shyu. Relevance feedback decision trees in content-based image retrieval. In *Content-based Access of Image and Video Libraries, 2000. Proceedings. IEEE Workshop on*, pages 68–72. IEEE, 2000.

[62] Heikki Mannila. Local and global methods in data mining: Basic techniques and open problems. In *Automata, Languages and Programming*, pages 57–68. Springer, 2002.

[63] N. Nanas, V. Uren, and A. De Roeck. A comparative evaluation of term weighting methods for information filtering. In *Database and Expert Systems Applications, 2004. Proceedings. 15th International Workshop on*, pages 13–17. IEEE, 2004.

[64] R.T. Ng, L.V.S. Lakshmanan, J. Han, and A. Pang. Exploratory mining and pruning optimizations of constrained associations rules. In *ACM SIGMOD Record*, volume 27, pages 13–24. ACM, 1998.

[65] S. Nijssen and J.N. Kok. Frequent graph mining and its application to molecular databases. In *Systems, Man and Cybernetics, 2004 IEEE International Conference on*, volume 5, pages 4571–4577. IEEE, 2004.

[66] E.R. Omiecinski. Alternative interest measures for mining associations in databases. *Knowledge and Data Engineering, IEEE Transactions on*, 15 (1):57–69, 2003.

[67] F. Pan, G. Cong, A.K.H. Tung, J. Yang, and M.J. Zaki. Carpenter: Finding closed patterns in long biological datasets. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 637–642. ACM, 2003.

[68] N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal. Discovering frequent closed itemsets for association rules. *Database TheoryICDT99*, pages 398–416, 1999.

[69] J. Pei and J. Han. Can we push more constraints into frequent pattern mining? In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 350–354. ACM, 2000.

[70] J. Pei, J. Han, R. Mao, et al. Closet: An efficient algorithm for mining frequent closed itemsets. In *ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, pages 21–30, 2000.

[71] L. Pipanmaekaporn and Y. Li. Mining a data reasoning model for personalized text classification. *IEEE Intelligent Informatics Bulletin*, 12(1): 17–24, 2011.

[72] K. Ramamohanarao and H. Fan. Patterns based classifiers. *World Wide Web*, 10(1):71–83, 2007.

[73] Marcela Xavier Ribeiro, Joselene Marques, Agma J. M. Traina, and Caetano Traina Jr. Statistical association rules and relevance feedback: Powerful allies to improve the retrieval of medical images. In *Proceedings of the 19th IEEE Symposium on Computer-Based Medical Systems*, pages 887–892, Washington, DC, USA, 2006. IEEE Computer Society. ISBN 0-7695-2517-1. doi: 10.1109/CBMS.2006.148. URL `http://dl.acm.org/citation.cfm?id=1152999.1153099`.

[74] S.E. Robertson and K.S. Jones. Relevance weighting of search terms. *Journal of the American Society for Information science*, 27(3):129–146, 2007.

[75] S.E. Robertson, S. Walker, and M. Beaulieu. Experimentation as a way of life: Okapi at trec. *Information Processing & Management*, 36(1):95–108, 2000.

[76] S.E. Robertson, S. Walker, H. Zaragoza, and R. Herbrich. Microsoft cambridge at TREC 2002: Filtering track. *NIST SPECIAL PUBLICATION SP*, pages 439–446, 2003.

[77] Y. Rui, T.S. Huang, S. Mehrotra, and M. Ortega. A relevance feedback architecture for content-based multimedia information retrieval systems. In *Content-Based Access of Image and Video Libraries, 1997. Proceedings. IEEE Workshop on*, pages 82–89. IEEE, 1997.

[78] I. Ruthven and M. Lalmas. A survey on the use of relevance feedback for information access systems. *The Knowledge Engineering Review*, 18(02): 95–145, 2003.

[79] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523, 1988.

[80] Gerard Salton, Anita Wong, and Chung-Shu Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975.

[81] Mark Sanderson and W. Bruce Croft. The history of information retrieval research. *Proceedings of the IEEE*, 100(Centennial-Issue):1444–1451, 2012.

[82] Silvia Schiaffino and Analía Amandi. Intelligent user profiling. In *Artificial Intelligence An International Perspective*, pages 193–216. Springer, 2009.

[83] F. Sebastiani. Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, 34(1):1–47, 2002.

[84] A. Silberschatz and A. Tuzhilin. What makes patterns interesting in knowledge discovery systems. *Knowledge and Data Engineering, IEEE Transactions on*, 8(6):970–974, 1996.

[85] R. Srikant and R. Agrawal. Mining sequential patterns: Generalizations and performance improvements. *Advances in Database TechnologyEDBT'96*, pages 1–17, 1996.

[86] P.N. Tan, V. Kumar, and J. Srivastava. Selecting the right interestingness measure for association patterns. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, page 41. ACM, 2002.

[87] R. Tesar, V. Strnad, K. Jezek, and M. Poesio. Extending the single words-based document model: a comparison of bigrams and 2-itemsets. In *Proceedings of the 2006 ACM symposium on Document engineering*, pages 138–146. ACM, 2006.

[88] J. Van Hulse and T. Khoshgoftaar. Knowledge discovery from imbalanced and noisy data. *Data & Knowledge Engineering*, 68(12):1513–1542, 2009.

[89] C. Wang and S. Parthasarathy. Summarizing itemset patterns using probabilistic models. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 730–735. ACM, 2006.

[90] C. Wang, K. Bi, Y. Hu, H. Li, and G. Cao. Extracting search-focused key n-grams for relevance ranking in web search. In *Proceedings of the fifth ACM international conference on Web search and data mining*, pages 343–352. ACM, 2012.

[91] J. Wang and G. Karypis. Harmony: Efficiently mining the best rules for classification. In *Proc. of SDM*, pages 205–216, 2005.

[92] J. Wang, J. Han, and J. Pei. Closet+: Searching for the best strategies for mining frequent closed itemsets. In *Proceedings of the ninth ACM SIGKDD*

*international conference on Knowledge discovery and data mining*, pages 236–245. ACM, 2003.

[93] X. Wang, H. Fang, and C.X. Zhai. A study of methods for negative relevance feedback. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 219–226. ACM, 2008.

[94] P. Willett. The porter stemming algorithm: then and now. *Program: electronic library and information systems*, 40(3):219–223, 2006.

[95] I.H. Witten, G.W. Paynter, E. Frank, C. Gutwin, and C.G. Nevill-Manning. Kea: Practical automatic keyphrase extraction. In *Proceedings of the fourth ACM conference on Digital libraries*, pages 254–255. ACM, 1999.

[96] S. K. Michael Wong and Yiyu Yao. Query formulation in linear retrieval models. *JASIS*, 41(5):334–341, 1990.

[97] S. K. Michael Wong, Yiyu Yao, and Peter Bollmann. Linear structure in information retrieval. In *Proceedings of the 11th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 219–232. ACM, 1988.

[98] S.T. Wu, Y. Li, Y. Xu, B. Pham, and P. Chen. Automatic pattern-taxonomy extraction for web mining. In *Proceedings of the 2004 IEEE/WIC/ACM International Conference on Web Intelligence*, pages 242–248. IEEE Computer Society Washington, DC, USA, 2004.

[99] S.T. Wu, Y. Li, and Y. Xu. Deploying approaches for pattern refinement in text mining. In *Proceedings of the Sixth International Conference on Data Mining*, pages 1157–1161. IEEE Computer Society, 2006.

[100] X. Wu, V. Kumar, J. Ross Quinlan, J. Ghosh, Q. Yang, H. Motoda, G.J. McLachlan, A. Ng, B. Liu, P.S. Yu, et al. Top 10 algorithms in data mining. *Knowledge and Information Systems*, 14(1):1–37, 2008.

[101] D. Xin, J. Han, X. Yan, and H. Cheng. Mining compressed frequent-pattern sets. In *Proceedings of the 31st international conference on Very large data bases*, page 720. VLDB Endowment, 2005.

[102] D. Xin, H. Cheng, X. Yan, and J. Han. Extracting redundancy-aware top-k patterns. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, page 453. ACM, 2006.

[103] Q. Xing, Y. Zhang, and L. Zhang. On bias problem in relevance feedback. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 1965–1968. ACM, 2011.

[104] H. Xiong, G. Pandey, M. Steinbach, and V. Kumar. Enhancing data analysis with noise removal. *Knowledge and Data Engineering, IEEE Transactions on*, 18(3):304–319, 2006.

[105] X. Yan, J. Han, and R. Afshar. Clospan: Mining closed sequential patterns in large datasets. In *Proceedings of SIAM International Conference on Data Mining*, pages 166–177, 2003.

[106] X. Yan, H. Cheng, J. Han, and D. Xin. Summarizing itemset patterns: a profile-based approach. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, page 323. ACM, 2005.

[107] Y. Yang. An evaluation of statistical approaches to text categorization. *Information retrieval*, 1(1):69–90, 1999.

[108] M.J. Zaki. Generating non-redundant association rules. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 34–43. ACM, 2000.

[109] M.J. Zaki. Spade: An efficient algorithm for mining frequent sequences. *Machine Learning*, 42(1):31–60, 2001.

[110] M.J. Zaki and C.J. Hsiao. Charm: An efficient algorithm for closed association rule mining. In *2nd SIAM International Conf. on Data Mining*, pages 457–473. Citeseer, 1999.

[111] M.J. Zaki et al. Scalable algorithms for association mining. *IEEE Transactions on Knowledge and Data Engineering*, 12(3):372–390, 2000.

[112] N. Zhong, Y. Li, and S. Wu. Effective pattern discovery for text mining. *Knowledge and Data Engineering, IEEE Transactions on*, (99):1–1, 2012.

[113] X. Zhou, Y. Li, P. Bruza, Y. Xu, and R.Y.K. Lau. Two-stage model for information filtering. In *Web Intelligence and Intelligent Agent Technology, 2008. WI-IAT'08. IEEE/WIC/ACM International Conference on*, volume 3, pages 685–689. IEEE, 2008.

[114] F. Zhu, X. Yan, J. Han, and P. Yu. gprune: a constraint pushing framework for graph pattern mining. *Advances in Knowledge Discovery and Data Mining*, pages 388–400, 2007.

[115] A. Zimmermann and L. De Raedt. Corclass: Correlated association rule mining for classification. In *Discovery Science*, pages 60–72. Springer, 2004.